symmetry

MDPI

Article

Improving Multivariate Microaggregation through Hamiltonian Paths and Optimal Univariate Microaggregation

Armando Maya-López [1], Fran Casino [2] and Agusti Solanas [1,*]

[1] Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili, Av. Països Catalans 26, 43007 Tarragona, Catalonia, Spain; armando.maya@estudiants.urv.cat
[2] Department of Informatics, University of Piraeus, Karaoli kai dimitriou 80, 18534 Pireas, Greece; francasino@unipi.gr
* Correspondence: agusti.solanas@urv.cat

Abstract: The collection of personal data is exponentially growing and, as a result, individual privacy is endangered accordingly. With the aim to lessen privacy risks whilst maintaining high degrees of data utility, a variety of techniques have been proposed, being microaggregation a very popular one. Microaggregation is a family of perturbation methods, in which its principle is to aggregate personal data records (i.e., microdata) in groups so as to preserve privacy through *k*-anonymity. The multivariate microaggregation problem is known to be NP-Hard; however, its univariate version could be optimally solved in polynomial time using the Hansen-Mukherjee (HM) algorithm. In this article, we propose a heuristic solution to the multivariate microaggregation problem inspired by the Traveling Salesman Problem (TSP) and the optimal univariate microaggregation solution. Given a multivariate dataset, first, we apply a TSP-tour construction heuristic to generate a Hamiltonian path through all dataset records. Next, we use the order provided by this Hamiltonian path (i.e., a given permutation of the records) as input to the Hansen-Mukherjee algorithm, virtually transforming it into a multivariate microaggregation solver we call Multivariate Hansen-Mukherjee (MHM). Our intuition is that good solutions to the TSP would yield Hamiltonian paths allowing the Hansen-Mukherjee algorithm to find good solutions to the multivariate microaggregation problem. We have tested our method with well-known benchmark datasets. Moreover, with the aim to show the usefulness of our approach to protecting location privacy, we have tested our solution with real-life trajectories datasets, too. We have compared the results of our algorithm with those of the best performing solutions, and we show that our proposal reduces the information loss resulting from the microaggregation. Overall, results suggest that transforming the multivariate microaggregation problem into its univariate counterpart by ordering microdata records with a proper Hamiltonian path and applying an optimal univariate solution leads to a reduction of the perturbation error whilst keeping the same privacy guarantees.

Keywords: microaggregation; statistical disclosure control; graph theory; traveling salesman problem; data privacy; location privacy

Citation: Maya-López, A.; Casino, F.; Solanas, A. Improving Multivariate Microaggregation through Hamiltonian Paths and Optimal Univariate Microaggregation. *Symmetry* **2021**, *13*, 916. https://doi.org/10.3390/sym13060916

Academic Editor: Egon Schulte

Received: 1 April 2021
Accepted: 14 May 2021
Published: 21 May 2021

1. Introduction

Knowledge retrieval and data processing are catalysts for innovation. The continuous advances in information and communication technologies (ICT) and the efficient processing of data allow the extraction of new knowledge by discovering non-obvious patterns and correlations in the data. Nevertheless, such knowledge extraction procedures may threaten individuals' privacy if the proper measures are not implemented to protect it [1–3]. For instance, an attacker may use publicly available datasets to obtain insights about individuals and extract knowledge by exploiting correlations that were not obvious from examining a single dataset [4]. Therefore, before disclosing any data, privacy protection procedures (e.g., anonymization, pseudonymization, aggregation, generalization) must be

applied. A wide variety of privacy models and protection mechanisms have been proposed in the literature so as to guarantee anonymity (at different levels depending on the utilized model) when disclosing data [5]. Since most privacy protection methods are based on modifying/perturbing/deleting original data, their main drawback is that they negatively affect the utility of the data. Hence, there is a need for finding a proper trade-off between data utility and privacy.

One of the most well-known disciplines studying methods to protect individuals' private information is Statistical Disclosure Control (SDC [6]), which seeks to anonymize microdata sets (i.e., datasets consisting of multiple records corresponding to individual respondents) in a way that it is not possible to re-identify the respondent corresponding to any particular record in the published microdata set. Microaggregation [7], which perturbs microdata sets by aggregating the attributes' values of groups of k records so as to reduce re-identification risk by achieving $k-$anonymity, stands out among the most widely used families of SDC methods. It is usually applied by statistical agencies to limit the disclosure of sensitive microdata, and it has been used to protect data in a variety of fields, namely healthcare [8], smart cities [9], or collaborative filtering applications [10], to name a few.

Although the univariate microaggregation problem can be optimally solved in polynomial time, optimal multivariate microaggregation is an NP-hard problem [11]. Thus, finding a solution for the multivariate problem requires heuristic approaches that aim to minimize the amount of data distortion (often measured in terms of information loss), whilst guaranteeing a desired privacy level (typically determined by a parameter k that defines the cardinality of the aggregated groups).

1.1. Contribution and Research Questions

In this article, we propose a novel solution for the multivariate microaggregation problem, inspired by the heuristic solutions of the Traveling Salesman Problem (TSP) and the use of the optimal univariate microaggregation algorithm of Hansen and Mukherjee (HM) [12]. Given an ordered numerical vector, the HM algorithm creates the optimal k-partition (i.e., the optimal univariate microaggregation solution). Hence, our intuition is that, if we feed the HM algorithm with a good ordering of the records in a multivariate dataset, it would output a good k-partition of the multivariate dataset (although not necessarily optimal).

Ordering the records of a univariate dataset is trivial. However, ordering those records in a multivariate dataset, in which every record has p attributes, is not obvious since it is not apparent how to determine the precedence of an element over another. Thus, the primary question is:

Q1: *How to create this ordering, when the records are in* $\mathbb{R}^p$.

We suggest that a possible order for the records in $\mathbb{R}^p$ is determined by the Hamiltonian path resulting from solving the Traveling Salesman Problem, in which the goal is to find the path that travels through all elements of a set only once, whilst minimizing the total length of the path. Optimally solving the TSP is known to be NP-Hard, but very good heuristic solutions are available. Hence, our intuition is that good heuristic solutions of the TSP (i.e., those with shorter path lengths) would provide a Hamiltonian path, that could be used as an ordered vector for the HM optimal univariate microaggregation algorithm, resulting in a good multivariate microaggregation solution.

The quality of a TSP solution is measured in terms of "path length", the shorter the length the better the solution. However, the quality of the microaggregation is measured in terms of information loss. Given a cardinality parameter k (which sets the minimum size of the aggregation clusters), the lower the information loss, the better the microaggregation. Hence, the next questions that we aim to answer are:

Q2: *Are the length of the Hamiltonian path and the information loss of the microaggregation related?, or Do shorter Hamiltonian paths lead to microaggregation solutions with lower information loss?*

and

Q3: *Is the length of the Hamiltonian path the only factor affecting information loss or does the particular construction of the path (regardless of the length) affect the information loss?*

Overall, the key question is:

Q4: *Does this approach provide better solutions (in terms of information loss) than the best performing microaggregation methods in the literature?*

In order to answer these questions, we have tested seven TSP solvers, combined with the HM algorithm (virtually applied in a multivariate manner, or Multivariate HM (MHM)). Particularly, we have tested the "Concorde" heuristic, which, to the best of our knowledge, is the first time it is used for microaggregation. In addition, we have tested well-known classic microaggregation methods (i.e., MDAV and V-MDAV), and an advanced refinement of the former (i.e., MDAV-LK-MHM).

With the aim to test all the aforementioned approaches on a variety of datasets, we have used three classical benchmarks (i.e., Census, Tarragona, and EIA) and three novel datasets containing trajectory data retrieved from public sources, which lead to our last research question:

Q5: *Do TSP-based microaggregation methods perform better than current solutions on trajectories datasets?*

1.2. Plan of the Article

The rest of the article aims to answer the research questions above, and it is organized as follows: Section 2 provides the reader with some fundamental knowledge on Statistical Disclosure Control and microaggregation. In addition, it introduces the basics of the Traveling Salesman Problem and an overview of the existing heuristics to solve it. Next, Section 3 analyzes related work and highlights the novelty of our proposal compared with the state of the art. Section 4 describes our proposal, which is later thoroughly tested and compared with well-known classical and state-of-the-art microaggregation methods in Section 5. Section 6 discusses the research questions and the main benefits of our proposal. The article concludes in Section 7 with some final remarks and comments on future research lines.

2. Background

2.1. Statistical Disclosure Control and Microaggregation

Statistical disclosure control (SDC) has the goal of preserving the statistical properties of datasets, whilst minimizing the privacy risks related to the disclosure of confidential information from individual respondents. Microaggregation is a family of SDC methods for microdata, which use data perturbation as a protection strategy.

Given an original data file D and a privacy parameter k, microaggregation can be defined as follows: Let us assume a microdata set D with p continuous numerical attributes and n records. Clusters (also referred to as groups or subsets in this context) of D are formed with n_i records in the i-th cluster ($n_i \geq k$ and $n = \sum_{i=1}^{g} n_i$), where g is the number of resulting clusters, and k a cardinality constraint. Optimal microaggregation is defined as the one yielding a k-partition maximizing the within-clusters homogeneity. Optimal microaggregation requires heuristic approaches since it is an NP-hard problem [11] for multivariate data. Microaggregation heuristics can be classified into two main families:

- Fixed-size microaggregation: These heuristics cluster the elements of D into k-partitions where all clusters have size k, except perhaps one group which has a size between k and $2k-1$, when the total number of records is not divisible by k.
- Variable-size microaggregation: These heuristics cluster the elements of D into k-partitions where all clusters have sizes in $(k, 2k-1)$. Note that it is easy to show that any cluster with size larger than $(2k-1)$ could be divided in several smaller clusters

of size between k and $2k-1$ in which its overall within-cluster homogeneity is better than that of the single larger cluster.

Therefore, a microaggregation process consists in constructing a k-partition of the dataset, this is a set of disjoint clusters (in which the cardinality is between k and $2k-1$) and replacing each original data record by the centroid (i.e., the average vector) of the cluster to which it belongs, hence creating a k-anonymous dataset D'. With the aim to reduce the information loss caused by the aggregation, the clusters are created so that the records in each cluster are similar.

2.2. Data Utility and Information Loss

The sum of square error (SSE) is commonly used for measuring the homogeneity in each group. In terms of sums of squares, maximizing within-groups homogeneity is equivalent to finding a k-partition minimizing the within-groups sum of square error (SSE) [13] defined as:

$$SSE = \sum_{i=1}^{g} \sum_{j=1}^{n_i} (x_{i,j} - \bar{x}_i)(x_{i,j} - \bar{x}_i)', \tag{1}$$

where $x_{i,j}$ is the j-th record in group i, and $\bar{x}_i$ is the average record of group i. The total sum of squares (SST), an upper bound on the partitioning information loss, can be computed as follows:

$$SST = \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})', \tag{2}$$

where x_i is the i-th record in D, and $\bar{x}_i$ is the average record of D. Note that all the above equations use vector notation, so $x_i \in \mathbb{R}^p$.

The microaggregation problem consists in finding a k-partition with minimum SSE, this is, the set of disjoint subsets of D so that $D = \bigcup_{m=1}^{g} s_m$, where s_m is the m-th subset, and g is the number of subsets, with minimum SSE. However, a normalized measure of information loss (expressed in percentage) is also used:

$$I_{loss} = \frac{SSE}{SST} \times 100. \tag{3}$$

In terms of information loss, the worst case scenario for microaggregation would happen when all records in D are replaced in D' by the average of the dataset (i.e., $SSE = SST \rightarrow I_{loss} = 100$), and the best case scenario implies that $D = D'$ (i.e., $k = 1$, no aggregation), which leads to $SSE = I_{loss} = 0$. Obviously, the latter case is optimal in terms of information loss, but it offers no privacy protection, at all. Hence, values for the protection parameter k are greater than one, typically: $k = 3, 4, 5$, or 6, and are chosen by privacy experts in statistical agencies so as to adapt to the needs of each particular dataset.

2.3. Basics on the Traveling Salesman Problem

The Traveling Salesman Problem (TSP) [14] consists of finding a particular *Hamiltonian cycle*. The problem can be stated as follows: a salesman leaves from one city and wants to visit (exactly once) each other city in a given group and, finally, return to the starting city. The salesman wonders in what order he should visit these cities so as to travel the shortest possible total distance.

In terms of graph theory, the TSP can be modeled by a graph $G = (V, E)$, where cities are the nodes in set $V = \{v_1, v_2, ..., v_n\}$ and each edge $e_{ij} \in E$ has an associated weight w_{ij} representing the distance between nodes i and j. The goal is to find a *Hamiltonian cycle*, i.e., a cycle which visits each node in the graph exactly once, with the least total weight. An alternative approach to the *Hamiltonian cycle* to solve the TSP is finding the *Shortest Hamiltonian path* through a graph (i.e., a path which visits each node in the graph exactly once). As an example, Figure 1 shows a short Hamiltonian path for the *Eurodist* dataset, which contains the distance (in km) between 21 cities in Europe.

Figure 1. A Hamiltonian path for the Eurodist dataset.

Finding an optimal solution to the TSP is known to be NP-Hard. Hence, several heuristics to find good but sub-optimal solutions have been developed. TSP heuristics typically fall into two groups: those involving minimum spanning trees for tour construction and those with edge exchanges to improve existing tours. There are numerous heuristics to solve the TSP [15,16]. In this article, we have selected a representative sample of heuristics, including well-known approaches and top performers from the state-of-the-art:

- **Nearest Neighbor algorithm**: The algorithm starts with a tour containing a randomly chosen node and appends the next nearest node iteratively.
- **Repetitive Nearest Neighbor**: The algorithm is an extension of the Nearest Neighbor algorithm. In this case, the tour is computed n times, each one considering a different starting node and then selecting the best tour as the outcome.
- Insertion Algorithms: All insertion algorithms start with a tour that originated from a random node. In each step, given two nodes already inserted in the tour, the heuristic selects a new node that minimizes the increase in the tour's length when inserted between such two nodes. Depending on the way such the next node is selected, one can find different variants of the algorithm. For instance, **Nearest Insertion**, **Farthest Insertion**, **Cheapest Insertion**, and **Arbitrary Insertion**.
- **Concorde**: This method is currently one of the best implementations for solving the symmetric TSP. It is based on the *Branch-and-Cut* method to search for optimal solutions.

3. Related Work on Microaggregation

There is a wide variety of heuristics to solve the multivariate microaggregation problem in the literature. One of the most well-known methods is the Maximum Distance to Average Vector (MDAV), proposed by Domingo-Ferrer et al. [17]. This method iteratively creates clusters of k members considering the furthest records from the dataset centroid. A variant of MDAV was proposed by Laszlo et al., namely the Centroid-Based Fixed Size method (CBFS) [18], which also has optimized versions based on kd-tree neighborhood search, such as KD-CBFS and KD-CBFSapp [19]. The Two Fixed Reference Points (TFRP) method was proposed by Chang et al. [20]. It uses the two most extreme points of the dataset at each iteration as references to create clusters. Differential Privacy-based microaggregation was explored by Yang et al. [21], which created a variant of the MDAV algorithm that uses the correlations between attributes to select the minimum required noise to achieve the desired privacy level. In addition, V-MDAV, a variable group-size heuristic based on the MDAV method was introduced by Solanas et al. in Reference [13]

with the aim to relax the cardinality constraints of fixed-size microaggregation and allow clusters to better adapt to the data and reduce the SSE.

Laszlo and Mukherjee [18] approached the microaggregation problem through minimum spanning trees, aimed at creating graph structures that can be pruned according to each node's associated weights to create the groups. Lin et al. proposed a Density-Based Algorithm (DBA) [22], which first forms groups of records in density descending order, and then fine-tunes these groups in reverse order. The successive Group Selection based on sequential Minimization of SSE (GSMS) method [23], proposed by Panagiotakis et al., optimizes the information loss by discarding the candidate cluster that minimizes the current SSE of the remaining records. Some methods are built upon the HM algorithm. For example, Mortazavi et al. proposed the IMHM method [24]. Domingo-Ferrer et al. [17] proposed a grouping heuristic that combines several methods, such as Nearest Point Next (NPN-MHM), MDAV-MHM, and CBFS-MHM.

Other approaches have focused on the efficiency of the microaggregation procedure, for example, the Fast Data-oriented Microaggregation (FDM) method proposed by Mortazavi et al. [25] efficiently anonymizes large multivariate numerical datasets for multiple successive values of k. The interested readers can find more detailed information about microaggregation in Reference [5,26].

The most similar work related to ours is the one presented in Reference [27] by Heaton and Mukherjee. The authors use TSP tour optimization heuristics (e.g., 2-opt, 3-opt) to refine a path created with the information of a multivariate microaggregation method (e.g., MDAV, MD, CBFS). Notice that, in our proposed method (described in the next section), we use tour construction TSP heuristics instead of optimization heuristics; thus, we eliminate the need for using a multivariate microaggregation method as a pre-processing step, and we decrease the computational time without hindering data utility.

4. Our Method

Our proposal is built upon two main building blocks: a TSP tour construction heuristic (H), and the optimal univariate microaggregation algorithm of Hansen and Mukherjee (HM). As we have already explained in Section 2, the HM algorithm is applied to univariate numerical data, because it requires the input elements to be in order. However, we virtually use it with multivariate data; thus, when we do that, we refer to it as Multivariate Hansen-Mukherjee (MHM), although, in practice, the algorithm is univariate. Since our proposal is based on a Heuristic (H) to obtain a Hamiltonian Path and the MHM algorithm, we have come to call it HMHM-microaggregation or $(HM)^2$-Micro, for short.

Given a multivariate microdata set (D) with p columns and r rows, we model it as a complete graph $G(N, E)$, where we assume that each row is represented by a node $n_i \in N$ (or a city, if we think in terms of the TSP), and each edge $e_{ij} \in E$ represents the Euclidean distance between n_i and n_j (or the distance between cities in TSP terms). Hence, we have a set of nodes $N = \{n_1, n_2, \dots n_r\}$ each representing rows of the microdata set in a multivariate space $\mathbb{R}^p$.

In a nutshell, we use H over G to create a Hamiltonian path (H_{path}) that travels across all nodes. H_{path} is a permutation ($\Pi^N = \{\pi_1^N, \pi_2^N, \dots \pi_r^N\}$) of the nodes in N, and *de facto* it determines an order for the nodes (i.e., it provides a sense of precedence between nodes). Hence, although D is multivariate, its rows represented as nodes in N can be sorted in a univariant permutation H_{path} that we use as input to the MHM algorithm. As a result, the MHM algorithm returns the optimal univariate k-partition of H_{path}, this is, the set of disjoint subsets $S = \{s_1, s_2, \dots s_t\}$ defining the clusters of N. Hence, since each node n_i represents a row in D, which is indeed multivariate, we have obtained a multivariate microaggregation of the rows in D and provided a solution for the multivariate microaggregation. Notice that, although MHM returns the optimal k-partition of H_{path}, it does not imply that the resulting microaggregation of D is optimal A schematic of our solution is depicted in Figure 2.

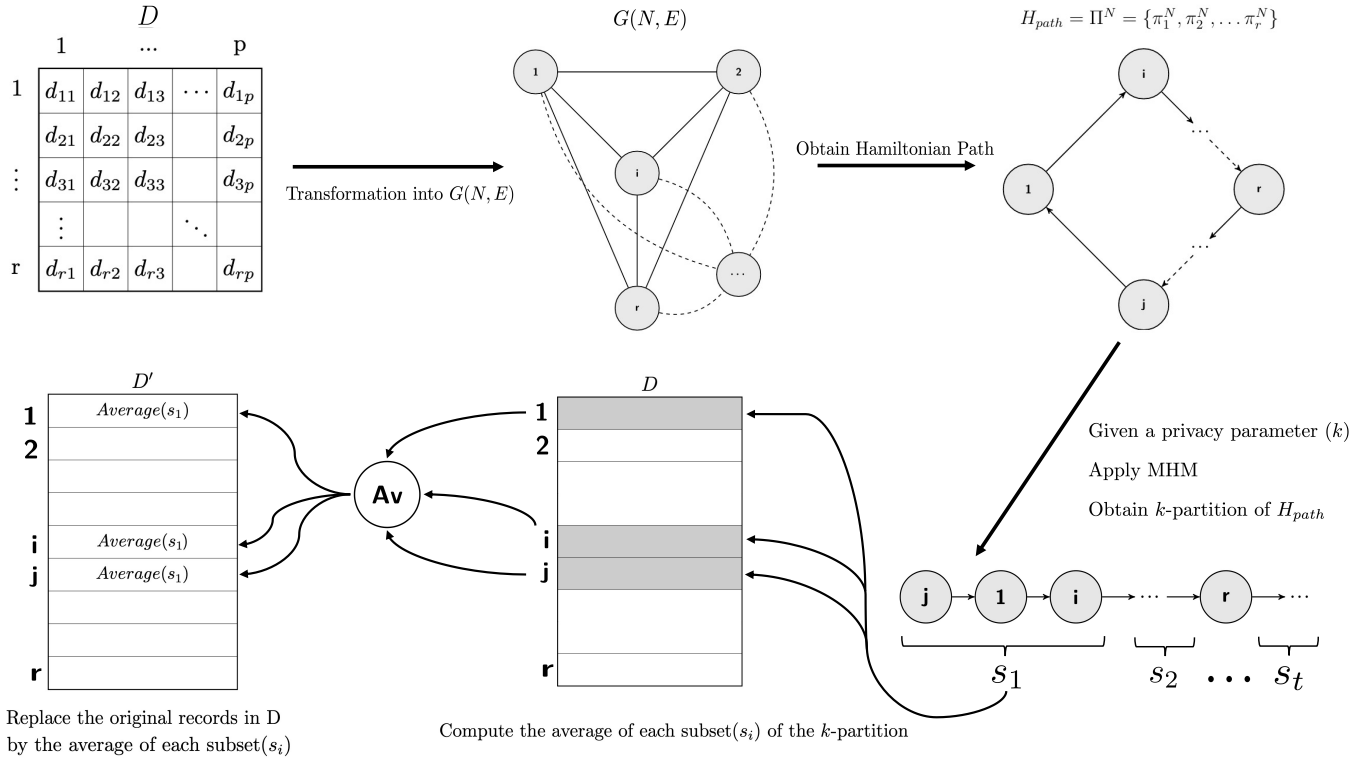

Figure 2. Given a microdata dataset, we use a tour construction heuristic to generate a Hamiltonian path, which will be used as the input of the MHM method to generate the groups.

Although the foundation of our proposal described above is pretty straightforward, it has the beauty of putting together complex mathematical building blocks from the multivariate and univariate worlds in a simple yet practical manner. In addition, our solution is very flexible, since it allows the use of any heuristic H to create the Hamiltonian path H_{path}, and it allows for comprehensive studies, such as the one we report in the next section.

Note that most TSP heuristics output a Hamiltonian cycle. However, since we need a Hamiltonian path, we use the well-known solution of adding a dummy node in the graph (i.e., a theoretical node in which its distance to all other nodes is zero), and we cut the cycle by eliminating this node, so as to obtain a Hamiltonian path.

For the sake of completeness, we summarize our proposal step-by-step in Algorithm 1, and we next comment on it. Our solution can be seen as a meta-heuristic to solve the multivariate microaggregation problem, since it can accommodate any Heuristic (H) able to create a Hamiltonian cycle from a complete graph (G), and it could deal with any privacy parameter (k). Thus, our algorithm receives as input a numerical multivariate microdata set D with p columns (attributes) and r rows, that have to be microaggregated, a Heuristic H, and a privacy parameter k (see Algorithm 1: line 1). In order to avoid bias towards higher magnitude variables, the original dataset D (understood as a matrix) is standardized by subtracting to each element the average of its column and dividing it by the standard deviation of the column. The result is a standardized dataset D_{std} in which each column has zero mean and unitary standard deviation (see Algorithm 1: line 2). Next, the distance matrix M_{dist} is computed. Each element $m_{ij} \in M_{dist}$ contains the Euclidean distance between row i and row j in D_{std}; hence, M_{dist} is a square matrix ($r \times r$) (see Algorithm 1: line 3). In order to be able to cut the Hamiltonian Cycle and obtain a Hamiltonian path, we add a dummy node to the dataset by adding a zero column and a zero row to M_{dist} and generate M^{dum}_{dist}, which is a square matrix ($r + 1 \times r + 1$) (see Algorithm 1: line 4). M^{dum}_{dist} is, in fact, a weighted adjacency matrix that defines a graph $G(N, E)$ with nodes $N = \{n_1, \ldots, n_{r+1}\}$ and edges $E = \{e_{11}, \ldots e_{i,j} \ldots e_{r+1,r+1}\} = \{M^{dum}_{dist\ 1,1}, \ldots M^{dum}_{dist\ r+1,r+1}\}$. With this matrix as

an input, we could compute a Hamiltonian Cycle H_{cycle} on G by applying a TSP heuristic H (see Algorithm 1: line 5). Notice that this Heuristic H could be anyone that gets as input a weighted graph and returns a Hamiltonian cycle. Some examples are: Concorde, Nearest Neighbor, Repetitive Nearest Neighbor, and Insertion Algorithms. After obtaining H_{cycle}, we cut it by removing the dummy node (see Algorithm 1: line 6), and we obtain a Hamiltonian path H_{path} that defines a permutation ($\Pi^N = \{\pi_1^N, \pi_2^N, \ldots \pi_r^N\}$) of the nodes in N, as well as determines an order for the nodes that can be inputted to the MHM algorithm to obtain its optimal k-partition (S) (see Algorithm 1: line 7). S is a set of disjoint subsets $S = \{s_1, s_2, \ldots s_t\}$ defining the clusters of nodes in N. Hence, with S and D, we could create a microaggregated dataset D' by replacing each row in D by the average vector of the k-partition subset to which it belongs (see Algorithm 1: line 8).

After applying the algorithm, we have transformed the original dataset D into a dataset D' that has been microaggregated so as to guarantee the privacy criteria established by k.

Algorithm 1 $(HM)^2$-Micro

1: **function** $(HM)^2$-MICRO(Microdata set D, TSP-Heuristic H, Privacy Parameter k)
2: D_{std} = StandardizeDataset(D)
3: M_{dist} = ComputeDistanceMatrix(D_{std})
4: M_{dist}^{dum} = InsertDummyNode(M_{dist})
5: H_{cycle} = CreateHamiltonianCycle(M_{dist}^{dum}, H)
6: H_{path} = CutDummyNode(H_{cycle})
7: S = MHM(H_{path}, D_{std}, k)
8: D' = BuildMicroaggregatedDataSet(D, S);
9: **return** D'
10: **end function**

5. Experiments

With the aim to practically validate the usefulness of our multivariate microaggregation proposal, we have thoroughly tested it on six datasets (described in Section 5.1) that serve as benchmarks. In addition, we are interested in knowing (if and) to what extend our method outperforms the best performing microaggregation methods in the literature. Hence, we have compared our proposal with these methods (described in Section 5.2), and the results of all these tests are summarized in Section 5.3. Overall, considering four different values for the privacy parameter $k \in \{3, 4, 5, 6\}$, ten microaggregation algorithms, 50 repetitions per case, and six datasets, we have run over 12.000 microaggregation tests, which allow us to provide a statistically solid set of results.

5.1. Datasets

We used six datasets as benchmarks for our experiments. We can classify those datasets into two main groups: The first group comprises three well-known SDC microdata sets that have been used for years as benchmarks in the literature, namely "Census", "EIA", and "Tarragona". The second group comprises three mobility datasets containing real GPS traces from three Spanish cities, namely "Barcelona", "Madrid", and "Tarraco". Notice that we use the term "Tarraco", the old Roman name for the city of Tarragona, in order to avoid confusion with the classic benchmark dataset "Tarragona". The features of each dataset are next summarized:

The Census dataset was obtained using the public *Data Extraction System of the U.S. Census Bureau*. It contains 1080 records with 13 numerical attributes. The Tarragona dataset was obtained from the Tarragona Chamber of Commerce. It contains information on 834 companies in the Tarragona area with 13 variables per record. The EIA dataset was obtained from the U.S. Energy Information Authority, and it consists of 4092 records with 15 attributes. More details on the aforementioned datasets can be obtained in Reference [28].

The Barcelona, Madrid, and Tarraco datasets consist of OpenStreetMap [29] GPS traces collected from those cities: Barcelona contains the GPS traces of the city of Barcelona within the area determined by the parallelogram formed by latitude (41.3726866, 41.4078446) and longitude (2.1268845, 2.1903992). The dataset has 969 records with 30 GPS locations each. Madrid contains the GPS traces of the city of Madrid within the area determined by the parallelogram formed by latitude (40.387613, 40.483515) and longitude (-3.7398145, -3.653985). The dataset has 959 records with 30 GPS locations each. Tarraco contains the GPS traces of the city of Tarragona within the area determined by the parallelogram formed by latitude (41.0967083, 41.141174) and longitude (1.226008, 1.2946691). The dataset has 932 records with 30 GPS locations each.

In all trajectories datasets, each record consists of 30 locations represented as (latitude and longitude). Hence, each record has 60 numerical values. These locations were extracted from each corresponding parallelogram according to the amount of recorded tracks and their length.

All datasets are available in our website: https://www.smarttechresearch.com/publications/symmetry2021-Maya-Casino-Solanas/ (accessed on 1 May 2021).

Table 1. Comparing methods and features. For Concorde, M is a bound on the time to explore subproblems, b is a branching factor, and d is a search depth.

Method		Cardinality	Computational Cost	Reference
MDAV		fixed	$O(n^2/2k)$	[17]
V-MDAV		variable	$O(n^2)$	[13]
MDAV-LK-MHM		variable	$O(n^2/2k)$	[27]
$(HM)^2$-Micro	**TSP Heuristic + MHM**			
	Nearest Neighbor	variable	$O(n^2)$	[15]
	Repetitive Nearest-Neighbor	variable	$O(n^2 \log n)$	[15]
	Nearest Insertion	variable	$O(n^2)$	[30]
	Farthest Insertion	variable	$O(n^2)$	[30]
	Cheapest Insertion	variable	$O(n^2)$	[30]
	Arbitrary Insertion	variable	$O(n^2)$	[30]
	Concorde	variable	$O(Mb^d)$	[31]

5.2. Compared Methods

We have selected a representative set of well-known and state-of-the-art methods to assess the value of our approach. We have selected two classic microaggregation methods (i.e., **MDAV** and **V-MDAV**), as baselines. In the case of V-MDAV, the method was run for several values of $\gamma \in \{0, 2\}$, and the best result is reported. Although some other newer methods might have achieved better results, they are still landmarks that deserve to be included in any microaggregation comparison.

For newer and more sophisticated methods, we have considered the work of Heaton and Mukherjee [27], in which they study a variety of microaggregation heuristics, including methods, such as CBFS and MD. Thus, instead of comparing our proposal with all those methods, we have taken the method that Heaton and Mukherjee reported as the best performer, namely the **MDAV-LK-MHM** method. This method, which is based on MDAV, first creates a microaggregation using MDAV, next improves the result of MDAV by apply-

ing the LK heuristic, and it finally applies MHM to obtain the resulting microaggregation (cf. Reference [27] for further details on the algorithm).

Regarding our proposal (i.e., $(HM)^2$**-Micro**), as we already discussed, it can be understood as a meta-heuristic able to embody any heuristic H that returns a Hamiltonian Cycle. Hence, with the aim to determine the best heuristic, we have analyzed seven alternatives, namely **Nearest Neighbor**, **Repetitive Nearest Neighbor**, **Nearest Insertion**, **Farther Insertion**, **Cheapest Insertion**, **Arbitrary Insertion**, and (our suggestion) **Concorde**. Table 1 summarizes some features of all selected methods, including the reference to the original article where the method was described. For our method, each reference points to the article describing the TSP heuristic.

The implementation of all these methods have used the *R* package *sdcMicro* [28], the TSP heuristics implemented in Reference [32], and the LK heuristics implemented in Reference [33]. LK has been configured so that the algorithm runs once at each iteration parameter `RUN`=1 until a local optimum is reached. This same criteria was followed for the other TSP heuristics. In this regard, the heuristics we used consider a random starting node at each run. Hence, each experiment has been repeated 50 times to guarantee statistically sound outcomes regardless of this random starting point.

5.3. Results Overview

By using the datasets and methods described above, we have analyzed the Information Loss (expressed in percentage), as a measure of data utility (cf. Section 2 for details). It is assumed that, given a privacy parameter k that guarantees that the microaggregated dataset is k-anonymous, the lower the Information Loss the better the result and performance of the microaggregation method. The results are reported in Tables 2–7 with the best (lowest) information loss highlighted in green.

Overall, it can be observed that our method, $(HM)^2$-Micro, with the Concorde heuristic is the best performer in 79% of the experiments, and it is the second best in the remaining 21% (for which the MDAV-LK-MHM outperforms it by a narrow margin of less than 2%). Interestingly enough, although $(HM)^2$-Micro, with both Nearest Insertion and Farthest-Insertion, is not the best performer in any experiment, it outperforms MDAV-LK-HMH 50% of the times. The rest of the methods obtain less consistent results and highly depend on the dataset.

When we analyze the results more closely for each particular dataset, we observe that, in the case of the "Census" dataset (cf. Table 2), our method with Concorde outperforms all methods for all values of k. In addition, despite the random nature of TSP-heuristics, the values of σ are very stable, denoting the robustness of all methods, yet slightly higher on average in the case of the methods with higher Information Loss. It is worth emphasizing though, that, in all runs, our method with Concorde and the MDAV-LK-MHM method obtained better results than MDAV and V-MDAV (i.e., the `max` values obtained in all runs are lower than the outcomes obtained by MDAV and V-MDAV).

Table 2. Information Loss obtained on the Census dataset.

	Census															
	$k=3$				$k=4$				$k=5$				$k=6$			
Method	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**
MDAV	5.6922	NA	NA	NA	7.4947	NA	NA	NA	9.0884	NA	NA	NA	10.3847	NA	NA	NA
V-MDAV	5.6619	NA	NA	NA	7.4947	NA	NA	NA	9.0070	NA	NA	NA	10.2666	NA	NA	NA
MDAV-LK-MHM	5.1085	0.0398	5.0256	5.1877	6.9131	0.0526	6.7774	7.0227	8.5199	0.0842	8.3100	8.7030	9.9752	0.1284	9.7675	10.2527
Nearest Insertion-MHM	5.6561	0.1369	5.3596	6.0695	7.4818	0.1579	7.1946	7.9318	8.9617	0.2539	8.5190	9.4727	10.3005	0.2927	9.7624	11.2086
Farthest Insertion-MHM	5.5638	0.0956	5.3300	5.8995	7.3485	0.0990	7.1723	7.5853	8.8234	0.1322	8.5784	9.1748	10.1250	0.1932	9.6970	10.7363
Cheapest Insertion-MHM	5.7044	0.0719	5.5669	5.8766	7.4625	0.1155	7.2674	7.8052	9.0340	0.1236	8.7212	9.3847	10.3787	0.1305	10.1706	10.9089
Arbitrary Insertion-MHM	5.5883	0.0976	5.4235	5.8763	7.3723	0.1438	7.1272	7.8250	8.8696	0.1788	8.5072	9.2867	10.2011	0.2475	9.7081	10.7794
Nearest Neighbor-MHM	6.9718	0.3508	6.1978	7.7291	9.2433	0.3702	8.6744	10.2246	11.3287	0.3854	10.5230	12.3958	13.1357	0.4053	12.4711	13.9421
Repetitive NN-MHM	6.2888	0.2192	5.8811	6.6841	8.6779	0.2799	7.9941	9.3345	10.7518	0.2472	10.3421	11.4554	12.5882	0.3143	11.9360	13.2915
Concorde-MHM	**5.0563**	0.0377	4.9917	5.1169	**6.8846**	0.0555	6.7895	7.0217	**8.4576**	0.0903	8.2372	8.6614	**9.8440**	0.1232	9.5542	10.2517

For the "EIA" dataset (cf. Table 3), MDAV-LK-MHM is the best performer for all values of k except $k = 5$, for which our proposal with Concorde performs better. In this case, the results obtained by these two methods are very close. Similarly to the results in "Census", the `max` values obtained by these two methods outperform MDAV and V-MDAV. In the case of "Tarragona" (cf. Table 4), our method with Concorde outperforms all other methods. Surprisingly, both MDAV and V-MDAV obtain better results than MDAV-LK-MHM, which performs poorly in this dataset.

Table 3. Information Loss obtained on the EIA dataset.

	EIA															
	$k = 3$				$k = 4$				$k = 5$				$k = 6$			
Method	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**
MDAV	0.4829	NA	NA	NA	0.6713	NA	NA	NA	1.6667	NA	NA	NA	1.3078	NA	NA	NA
V-MDAV	0.4829	NA	NA	NA	0.6713	NA	NA	NA	1.2771	NA	NA	NA	1.2320	NA	NA	NA
MDAV-LK-MHM	**0.3741**	0.0075	0.3659	0.4097	**0.5251**	0.0116	0.5117	0.5693	0.7890	0.0336	0.7502	0.8932	**1.0430**	0.0289	1.0033	1.1113
Nearest Insertion-MHM	0.4061	0.0114	0.3831	0.4238	0.5781	0.0241	0.5441	0.6179	0.8621	0.0456	0.8032	0.9760	1.1254	0.0837	0.9976	1.3334
Farthest Insertion-MHM	0.4070	0.0119	0.3872	0.4207	0.5878	0.0251	0.5524	0.6277	0.8764	0.0522	0.8190	0.9747	1.1776	0.0359	1.1245	1.2484
Cheapest Insertion-MHM	0.5254	0.0358	0.4692	0.5651	0.7321	0.0641	0.6322	0.8477	1.0868	0.0689	0.9910	1.2264	1.4061	0.1147	1.2605	1.6329
Arbitrary Insertion-MHM	0.4281	0.0300	0.3921	0.4944	0.6092	0.0376	0.5566	0.6699	0.9048	0.0840	0.8194	1.0621	1.1928	0.1077	1.0652	1.3476
Nearest Neighbor-MHM	0.9028	0.1455	0.5089	1.1023	1.1510	0.1675	0.7056	1.3776	1.4015	0.1788	0.9451	1.6767	1.6792	0.1107	1.4635	1.9139
Repetitive NN-MHM	0.5110	0.0532	0.4725	0.6599	0.7192	0.0557	0.6646	0.8619	1.0072	0.0701	0.9274	1.1126	1.3101	0.1521	1.1561	1.4825
Concorde-MHM	0.3889	0.0203	0.3673	0.4210	0.5288	0.0170	0.5087	0.5576	**0.7802**	0.0267	0.7581	0.8501	1.0476	0.0282	1.0009	1.0904

Table 4. Information Loss obtained on the Tarragona dataset.

	Tarragona															
	$k = 3$				$k = 4$				$k = 5$				$k = 6$			
Method	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**
MDAV	16.9326	NA	NA	NA	19.5460	NA	NA	NA	22.4619	NA	NA	NA	26.3252	NA	NA	NA
V-MDAV	16.6603	NA	NA	NA	19.5460	NA	NA	NA	22.4619	NA	NA	NA	26.3252	NA	NA	NA
MDAV-LK-MHM	18.7969	1.8738	15.0595	23.0830	22.8523	1.7576	19.1195	26.2806	26.2432	1.5066	23.0421	28.9522	28.5244	1.7742	25.1703	30.9656
Nearest Insertion-MHM	15.9687	0.8360	15.1107	20.1835	19.3677	1.3141	17.8032	24.5286	23.7323	1.4376	21.8365	28.9753	26.9018	1.5674	24.6538	33.0785
Farthest Insertion-MHM	15.7634	0.2062	15.4743	16.6623	19.0323	0.5521	18.1062	20.2105	22.8316	0.7636	21.3313	24.1988	25.7627	0.4496	24.9004	26.9613
Cheapest Insertion-MHM	16.3142	1.4861	15.2169	22.0271	19.7784	1.6060	18.3103	25.8916	23.9017	1.7155	22.3121	30.0828	27.5572	1.6611	25.2394	32.7082
Arbitrary Insertion-MHM	16.0918	0.7527	15.1310	18.9668	19.5461	1.3436	18.2072	25.8572	23.7685	1.3985	21.7333	29.1863	27.0419	1.6872	25.0093	33.2382
Nearest Neighbor-MHM	22.3019	0.8866	19.9620	23.5496	27.1002	1.2234	24.2527	29.5117	30.4478	1.5455	27.7026	33.3513	34.5445	1.2088	31.3302	37.5350
Repetitive NN-MHM	17.6981	1.2157	15.7435	20.9981	22.1232	1.9138	20.0839	28.7399	27.9089	1.7946	25.1434	32.5729	30.4085	1.9216	28.0648	35.2458
Concorde-MHM	**14.7677**	0.0858	14.6294	14.9633	**17.9957**	0.1241	17.7528	18.2211	**21.9895**	0.2164	21.6712	22.3479	**25.3459**	0.2061	24.8045	25.6564

So, it can be concluded that the overall winner for the classical benchmarks (i.e., Census, EIA, and Tarragona) is our method, $(HM)^2$-Micro, with the Concorde heuristic, that is only marginally outperformed by MDAV-LK-MHM in the EIA dataset.

Regarding the other three datasets containing GPS traces (i.e., Barcelona, Madrid and Tarraco), our method, $(HM)^2$-Micro, with the Concorde heuristic, is the best performer in 83% of the cases and comes second best in the remaining 17%. For the Barcelona dataset (cf. Table 5), MDAV-LK-MHM and $(HM)^2$-Micro, with the Concorde heuristic, perform very well and similarly. The methods with the worst Information Loss are MDAV and V-MDAV. Our method, $(HM)^2$-Micro, with the Insertion heuristics, have a remarkable performance, obtaining values similar to those of MDAV-LK-MHM and Concorde. Nevertheless, it is worth noting that the max (worst) values obtained by MDAV-LK-MHM and Concorde are still better than the averages obtained by the other methods. In the case of the Madrid dataset (cf. Table 6), our method, $(HM)^2$-Micro, with the Concorde heuristic, achieves the minimum (best) value of Information Loss for all values of k. We can also observe that our method with Insertion heuristics offers higher performance than MDAV-LK-MHM. Finally, the results for the Tarraco dataset (cf. Table 7) show that the minimum (best) Information Loss value is obtained by our method with the Concorde heuristic in all cases. In this case, MDAV-LK-MHM performs poorly, and, for $k = 3$ and $k = 4$, MDAV and V-MDAV are better.

Table 5. Information Loss obtained on the Barcelona dataset.

	Barcelona															
	$k = 3$				$k = 4$				$k = 5$				$k = 6$			
Method	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**
MDAV	2.5667	NA	NA	NA	3.5023	NA	NA	NA	4.2849	NA	NA	NA	5.1873	NA	NA	NA
V-MDAV	2.5667	NA	NA	NA	3.3193	NA	NA	NA	4.2849	NA	NA	NA	5.1873	NA	NA	NA
MDAV-LK-MHM	**1.6251**	0.0362	1.5637	1.7425	**2.1913**	0.0339	2.1170	2.2738	2.6798	0.0607	2.5156	2.8067	3.2120	0.0664	3.0731	3.3825
Nearest Insertion-MHM	1.8022	0.0656	1.6857	1.9438	2.3526	0.0842	2.1754	2.5050	2.8405	0.1008	2.6417	3.0411	3.3316	0.1083	3.1093	3.5103
Farthest Insertion-MHM	1.7838	0.0525	1.6967	1.8980	2.3575	0.0698	2.1919	2.4681	2.8386	0.0751	2.6654	2.9670	3.3189	0.1131	3.1112	3.6445
Cheapest Insertion-MHM	1.8156	0.0565	1.6887	1.9293	2.3880	0.0912	2.2354	2.5473	2.8887	0.0792	2.7807	3.0405	3.4118	0.1238	3.1938	3.6247
Arbitrary Insertion-MHM	1.8061	0.0635	1.6823	1.9469	2.3593	0.0749	2.1808	2.5414	2.8231	0.0911	2.6338	3.0251	3.3331	0.1085	3.1031	3.5584
Nearest Neighbor-MHM	2.2019	0.1202	1.9165	2.4476	2.9274	0.1778	2.5276	3.3377	3.4733	0.2168	3.0611	3.9399	4.1053	0.2590	3.5159	4.6420
Repetitive NN-MHM	2.0091	0.0563	1.8899	2.2547	2.7474	0.0611	2.6108	3.0130	3.2318	0.1001	3.1176	3.5701	3.8877	0.1220	3.7106	4.1982
Concorde-MHM	1.6829	0.0375	1.6210	1.7848	2.2132	0.0534	2.1138	2.3426	**2.6786**	0.0627	2.4974	2.8268	**3.1075**	0.0718	2.9588	3.2348

Table 6. Information Loss obtained on the Madrid dataset.

	Madrid															
	$k = 3$				$k = 4$				$k = 5$				$k = 6$			
Method	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**
MDAV	3.1876	NA	NA	NA	4.3353	NA	NA	NA	5.2883	NA	NA	NA	5.8235	NA	NA	NA
V-MDAV	3.1876	NA	NA	NA	4.3353	NA	NA	NA	5.2883	NA	NA	NA	5.8235	NA	NA	NA
MDAV-LK-MHM	2.9872	0.1285	2.7200	3.1946	4.0536	0.1398	3.6804	4.3314	4.8541	0.1664	4.4680	5.1856	5.5703	0.2163	5.0931	6.0088
Nearest Insertion-MHM	2.7511	0.0814	2.5782	2.9116	3.7039	0.1122	3.4304	3.9623	4.4522	0.1535	4.1533	4.8463	5.1544	0.1549	4.8661	5.5510
Farthest Insertion-MHM	2.6683	0.0558	2.5319	2.8280	3.6187	0.0742	3.4605	3.7755	4.3338	0.1131	4.1260	4.5668	5.0598	0.1172	4.8391	5.3372
Cheapest Insertion-MHM	2.7833	0.0749	2.6517	2.9789	3.7531	0.0804	3.5253	3.9830	4.4752	0.1140	4.3163	4.7356	5.2496	0.1345	5.0147	5.5609
Arbitrary Insertion-MHM	2.7476	0.0757	2.6009	2.9160	3.7156	0.0986	3.5213	3.9828	4.4149	0.1420	4.0583	4.7078	5.1070	0.1437	4.7687	5.3754
Nearest Neighbor-MHM	3.4257	0.1714	3.0816	3.9040	4.7553	0.2116	4.2823	5.3736	5.7671	0.2194	5.1807	6.3191	6.7615	0.2507	6.1871	7.4355
Repetitive NN-MHM	3.1236	0.1345	2.8799	3.5430	4.4141	0.1482	4.1254	5.0012	5.3911	0.2127	5.0894	6.1676	6.4865	0.2223	6.1764	7.3492
Concorde-MHM	**2.4845**	0.0336	2.4053	2.5728	**3.4302**	0.0466	3.3249	3.5664	**4.1124**	0.0774	3.9816	4.3228	**4.8066**	0.1065	4.6538	5.0534

Table 7. Information Loss obtained on the Tarraco dataset.

	Tarraco															
	$k = 3$				$k = 4$				$k = 5$				$k = 6$			
Method	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**	**Average**	σ	**min**	**max**
MDAV	0.9988	NA	NA	NA	1.4180	NA	NA	NA	1.7683	NA	NA	NA	2.0260	NA	NA	NA
V-MDAV	0.9988	NA	NA	NA	1.3093	NA	NA	NA	1.7182	NA	NA	NA	2.0051	NA	NA	NA
MDAV-LK-MHM	1.1365	0.0154	1.0979	1.1465	1.4216	0.0203	1.4115	1.4723	1.7201	0.0401	1.6995	1.8257	2.0238	0.0404	2.0061	2.1247
Nearest Insertion-MHM	0.9113	0.0345	0.8490	1.0100	1.2634	0.0745	1.1052	1.4306	1.5988	0.1160	1.4220	1.8839	1.9105	0.1517	1.7018	2.2870
Farthest Insertion-MHM	0.9190	0.0368	0.8582	1.0268	1.2217	0.0490	1.1123	1.3755	1.5040	0.0581	1.3965	1.7118	1.8346	0.0612	1.7533	2.1299
Cheapest Insertion-MHM	0.9500	0.0406	0.8975	1.0962	1.2951	0.0557	1.2270	1.4637	1.6200	0.0870	1.5225	1.8677	1.9704	0.1094	1.8584	2.2471
Arbitrary Insertion-MHM	0.9258	0.0455	0.8589	1.0269	1.2530	0.0753	1.1419	1.4538	1.5695	0.0971	1.4454	1.8312	1.9051	0.1265	1.7475	2.3396
Nearest Neighbor-MHM	1.5080	0.1937	1.1624	2.0189	2.1341	0.2232	1.5881	2.6725	2.6499	0.2671	2.0802	3.2271	3.3041	0.4123	2.6557	4.3884
Repetitive NN-MHM	1.2177	0.1286	1.0276	1.5906	1.7806	0.1599	1.4244	2.1131	2.2545	0.1882	1.9146	2.7394	2.7384	0.2209	2.3073	3.4314
Concorde-MHM	**0.8482**	0.0179	0.8167	0.9005	**1.1031**	0.0324	1.0739	1.2348	**1.3805**	0.0556	1.3275	1.6813	**1.7280**	0.0652	1.6610	2.1308

We have already discussed that all studied methods (with the exception of MDAV and V-MDAV) have a non-deterministic component emerging from the random selection of the initial node. This random selection affects the performance of the final microaggregation obtained. With the aim to analyze the effect of this non-deterministic behavior, we have studied the standard deviation of all methods for all values of k and for all datasets. In addition, we have visually inspected the variability of the results by using box plot diagrams.

Since the results are quite similar and consistent across all datasets, for the sake of clarity, we only reproduce here the box plots for the "Census" dataset (see Figure 3), and we leave the others in Appendix A for the interested reader.

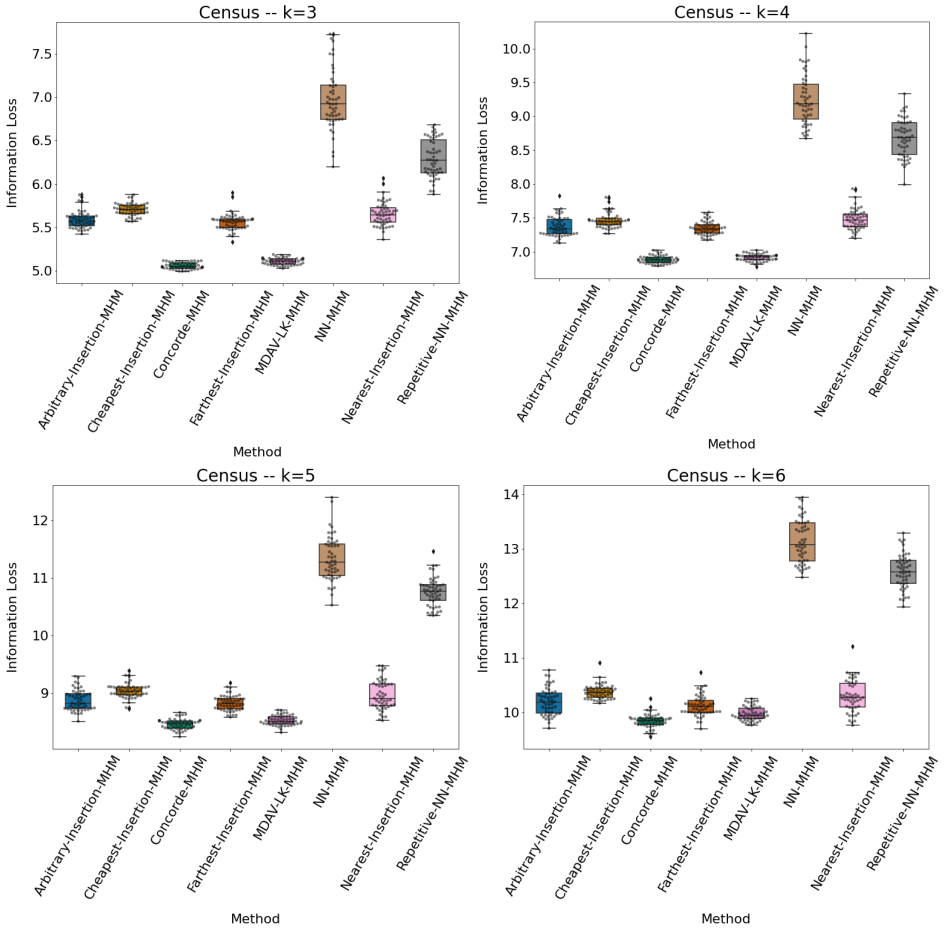

Figure 3. Information Loss variability for each value of k over the Census dataset.

In Figure 3, we can observe that the Information Loss values increase with k, but all methods have the same behavior regardless of the value of k. In addition, it is clear that the most stable methods are $(HM)^2$-Micro, with Concorde, and MDAV-LK-MHM.

Overall, we observe some expected differences depending on the datasets. However, the behavior of the best performing methods is stable. Particularly, the datasets with GPS traces (i.e., Barcelona, Madrid, and Tarraco) show more stable results. In summary, the best method was our $(HM)^2$-Micro with Concorde, exhibiting the most stable results across all datasets.

6. Discussion

Over the previous sections, we have presented our microaggregation method, $(HM)^2$-Micro, its rationale, and its performance against other classic and state-of-the-art methods on a variety of datasets. In the previous section, we have reported the main results, and we will discuss them next by progressively answering the research questions that we posed in the Introduction of the article.

> ***Q1:*** *How to create a suitable ordering for a univariate microaggregation algorithm, when the records are in* $\mathbb{R}^p$.

A main takeaway of this article is that, by using a combination of TSP tour construction heuristics (e.g., Concorde) and an optimal univariate microaggregation algorithm, we are properly ordering multivariate datasets in a univariate fashion that leads to excellent multivariate microaggregation solutions. Other approaches to order $\mathbb{R}^p$ points might consider projecting them over the principal component. However, the information loss associated with this approach makes it unsuitable. In addition, other more promising approaches, like the one used in MDAV-LK-MHM, first create a k-partition and set an

order based on maximum distance criteria. Although this approach might work well in some cases, we have clearly seen that Hamiltonian paths created by TSP-heuristics, like Concorde, outperform this approach. Hence, based on the experiments of Section 5, we can conclude that TSP-heuristics, like Concorde, provide an order for elements in $\mathbb{R}^p$ that is suitable for an optimal univariate microaggregation algorithm to output a consistent multivariate microaggregation solution with low Information Loss (i.e., high data utility). Moreover, from all analyzed heuristics, it is clear that the best performer is Concorde, followed by insertion heuristics.

> ***Q2:*** *Are the length of the Hamiltonian path and the information loss of the microaggregation related?, or Do shorter Hamiltonian paths lead to microaggregation solutions with lower information loss?*

When we started this research, our intuition was that good heuristic solutions of the TSP (i.e., those with shorter path lengths) would provide a Hamiltonian path, that could be used as an ordered vector for the HM optimal univariate microaggregation algorithm, resulting in a good multivariate microaggregation solution. From this intuition, we assumed that shorter Hamiltonian paths would lead to lower Information Loss in microaggregated datasets.

In order to validate (or disproof) this intuition, we have analyzed the Pearson correlation between the Hamiltonian path length obtained by all studied heuristics (i.e., Nearest Neighbor, Repetitive Nearest Neighbor, Nearest Insertion, Farther Insertion, Cheapest Insertion, Arbitrary Insertion, and Concorde) and the SSE of the resulting microaggregation. We have done so for all studied datasets and k values. The results are summarized in Table 8, and all plots along with a trend line are available in Appendix B.

Table 8. Summary of the Pearson correlation between Path Length and SSE.

Dataset	$k = 3$	$k = 4$	$k = 5$	$k = 6$
Census	0.48	0.39	0.32	0.28
EIA	0.62	0.67	0.74	0.76
Tarragona	0.70	0.72	0.82	0.71
Barcelona	0.83	0.81	0.81	0.80
Madrid	0.84	0.81	0.80	0.78
Tarraco	0.80	0.82	0.82	0.80

From the correlation analysis, it can be concluded that there is a positive correlation between the Hamiltonian path length and the SSE. This is, the shorter the path length the lower the SSE. This statement holds for all k and for all datasets (although Census exhibits a lower correlation). Hence, although this result is not a causality proof, it can be safely said that good solutions of the TSP problem lead to good solutions of the multivariate microaggregation problem. In fact, the best heuristic (i.e., Concorde) always results in the lowest (best) SSE.

Interested readers can find all plots in Appendix B. However, for the sake of clarity, let us illustrate this result by discussing the case of the Madrid dataset with $k = 6$, depicted in Figure 4. In the figure, the positive correlation is apparent. In addition, it is clear that heuristics tend to form clusters. In a nutshell, the best heuristic is Concorde, followed by the insertion family of methods (i.e., Nearest Insertion, Furthest Insertion, Cheapest Insertion, and Arbitrary Insertion), followed by Repetitive Nearest Neighbor and Nearest Neighbor.

Although Figure 4 clearly illustrates the positive correlation between the path length and the SSE, it also shows that heuristics tend to cluster and might indicate that not only the path but the heuristic (per se) plays a role in the reduction of the SSE. This indication leads us to our next research question.

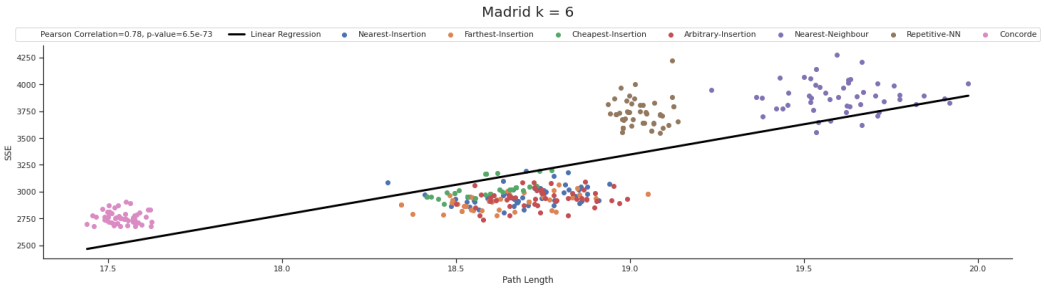

Figure 4. Relation between SSE and Path Length for Madrid and $k = 6$.

> ***Q3:*** *Is the length of the Hamiltonian path the only factor affecting information loss or does the particular construction of the path (regardless of the length) affect the information loss?*

In the previous question, we have found clear positive correlation between the path length and the SSE. However, we have also observed apparent clusters suggesting that the very heuristics could be responsible for the minimization of the SSE. In other words, although the path length and SSE are positively correlated when all methods are analyzed together, would this correlation hold when heuristics are analyzed one at a time? In order to answer this question, we have analyzed the results of each heuristic individually, and we have observed that there is still positive correlation between path length and SSE, but it is very weak or almost non-existent (i.e., very close to 0), as Figure 5 illustrates.

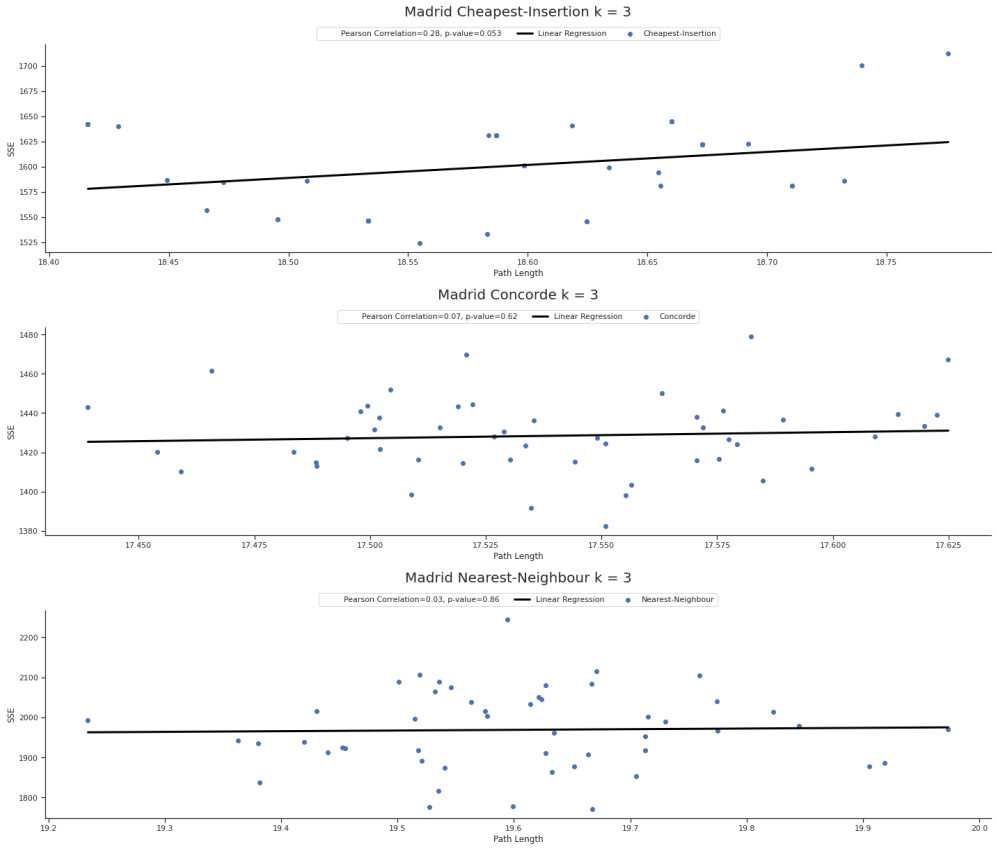

Figure 5. Correlation between path length and SSE for each individual method (from top to bottom: Cheapest Insertion, Concorde, and Nearest Neighbor) for $k = 3$ over the Madrid dataset.

The results shown in Figure 5 are only illustrative, and a deeper analysis that is out of the scope of this paper would be necessary. However, our initial results indicate that, although there is positive correlation between path length and SSE globally, this correlation

weakens significantly when analyzed on each heuristic individually. This result suggests that it is not only the length of the path but the way in which this path is constructed what affects the SSE. This would explain why similar methods (e.g., those based on insertion) behave similarly in terms of SSE although their paths' length varies.

Q4: *Does $(HM)^2$-Micro provide better solutions (in terms of information loss) than the best performing microaggregation methods in the literature?*

This question has been already answered in Section 5.3. However, for the sake of completeness, we summarize it here: The results obtained after executing more than 12,000 tests suggest that our solution $(HM)^2$-Micro obtains better results than classic microaggregation methods, such as MDAV and V-MDAV. Moreover, when $(HM)^2$-Micro uses the Concorde heuristic to determine the Hamiltonian path, it outperforms the best state-of-the-art methods consistently. In our experiments, $(HM)^2$-Micro with Concorde was the best performer 79% of the times and was the second best in the remaining 21%.

Q5: *Do TSP-based microaggregation methods perform better than current solutions on trajectories datasets?*

$(HM)^2$-Micro with Concorde is the best overall performer. Moreover, if we focus on those datasets with trajectory data (i.e., Barcelona, Madrid, and Tarraco), the results are even better. It is the best performer in 83% of the tests and the second best in the remaining 17%. This good behavior of the method could result from the very foundations of the TSP; however, there is still plenty of research to do in this line to reach more solid conclusions. Location privacy is a very complex topic that encompasses many nuances beyond *k*-anonymity models (such as the one followed in this article). However, this result is an invigorating first step towards the analysis of novel microaggregation methods applied to trajectory analysis and protection.

7. Conclusions

Microaggregation has been studied for decades now, and, although finding the optimal microaggregation is NP-Hard and a polynomial-time microaggregation algorithm has not been found, steady improvements over microaggregation heuristics have been made. Hence, after such a long research and polishing process, finding new solutions that improve the best methods is increasingly difficult. In this article, we have presented $(HM)^2$-Micro, a meta-heuristic that leverages the advances in TSP solvers and combines them with the optimal univariate microaggregation to create a flexible and robust multivariate microaggregation solution.

We have studied our method and thoroughly compared it to classic and state-of-the-art microaggregation algorithms over a variety of classic benchmarks and trajectories datasets. Overall, we have executed more than 12,000 tests, and we have shown that our solution embodying the Concorde heuristic outperforms the others. Hence, we have shown that our TSP-inspired method could be used to guarantee *k*-anonymity of trajectories datasets whilst reducing the Information Loss, thus increasing data utility. Furthermore, our proposal is very stable, i.e., it does not change significantly its performance regardless of the random behavior associated with initial nodes selection.

In addition to proposing $(HM)^2$-Micro, we have found clear correlations between the length of Hamiltonian Paths and the SSE introduced by microaggregation processes, and we have shown the importance of the Hamiltonian Cycle construction algorithms over the overall performance of microaggregation.

Despite these relevant results, there is still much to do in the study of microaggregation and data protection. Future work will focus on scaling up $(HM)^2$-Micro to high-dimensional and very-large datasets. Considering the growing importance of Big Data and Cloud Computing, adapting our solution to distributed computation environments is paramount. Moreover, adjusting TSP heuristics to leverage lightweight microaggregation-based approaches is an interesting research path to follow. In addition, although the values of the privacy parameter *k* are typically low (i.e., 3, 4, 5, 6), we plan to study the effect of

larger values of k on our solution. Last but not least, since microaggregation is essentially a data-oriented procedure, we will study how our solution adapts to data structures from specific domains, such as healthcare, transportation, energy, and the like.

All in all, with $(HM)^2$-Micro, we have set the ground for the study of multivariate microaggregation meta-heuristics from a new perspective, that might continue in the years to come.

Author Contributions: Conceptualization, A.M.-L. and A.S.; methodology, A.S.; software, A.M.-L.; validation, A.M.-L., F.C. and A.S.; formal analysis, A.S.; investigation, A.M.-L., F.C. and A.S.; resources, A.S.; data curation, A.M.-L.; writing—original draft preparation, A.M.-L. and A.S.; writing—review and editing, F.C. and A.S.; visualization, F.C.; supervision, F.C. and A.S.; project administration, A.S.; funding acquisition, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the European Commission under the Horizon 2020 Programme (H2020), as part of the project *LOCARD* (https://locard.eu (accessed on 1 May 2021)) (Grant Agreement no. 832735), and by the Spanish Ministry of Science & Technology with project IoTrain RTI2018-095499-B-C32. The content of this article does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the authors.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset and high resolution data can be found in: https://www.smarttechresearch.com/publications/symmetry2021-Maya-Casino-Solanas/ (accessed on 1 May 2021).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Information Loss Variability Box Plots

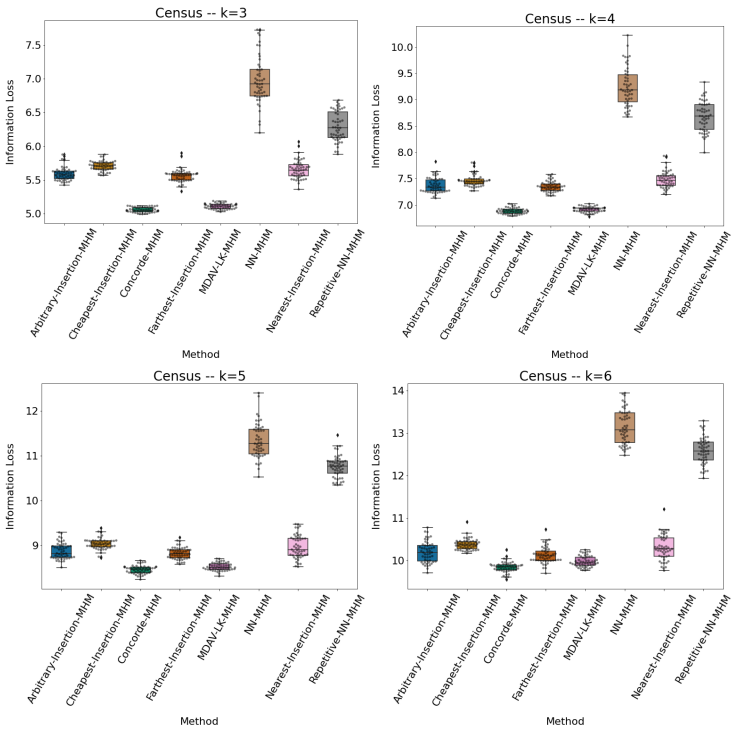

Figure A1. Information Loss variability for each value of k over the Census dataset.

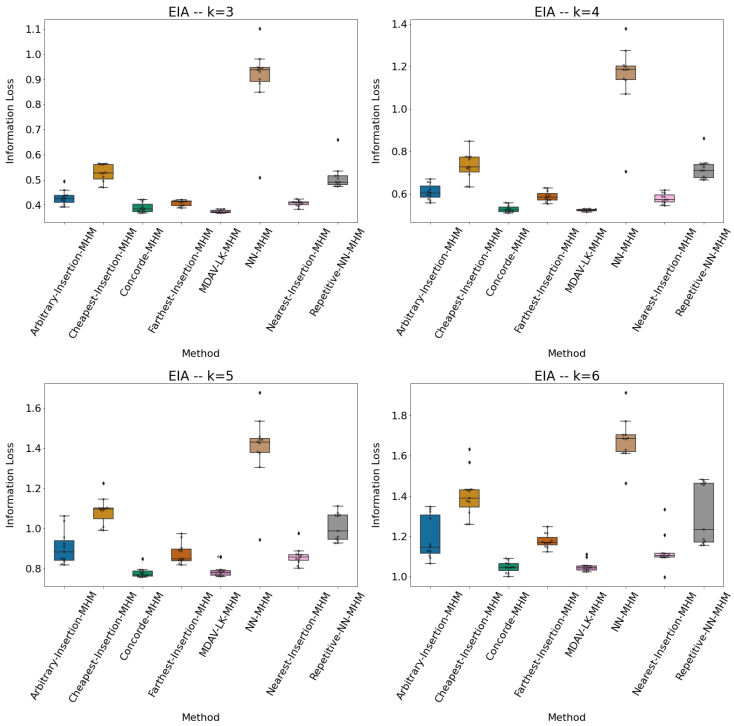

Figure A2. Information Loss variability for each value of k over the EIA dataset.

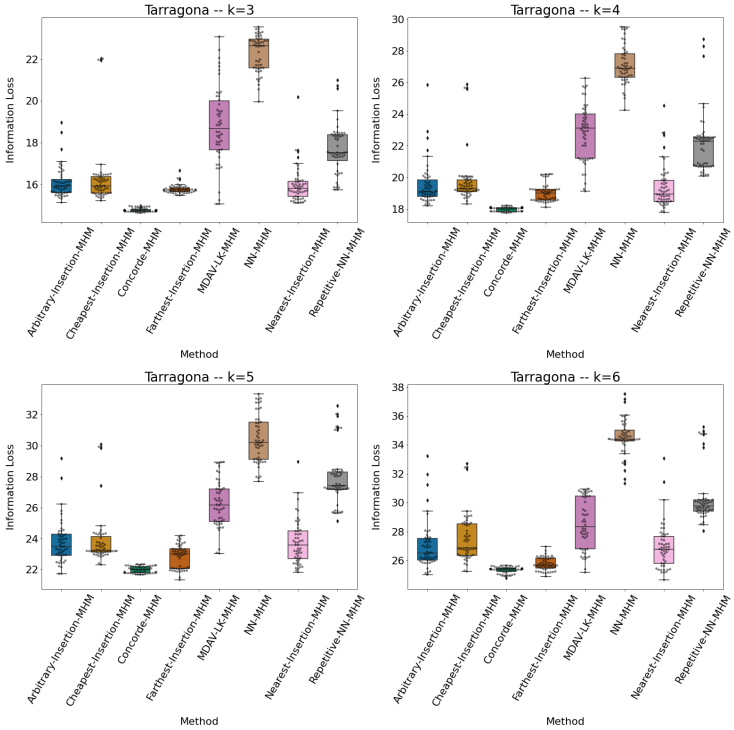

Figure A3. Information Loss variability for each value of k over the Tarragona dataset.

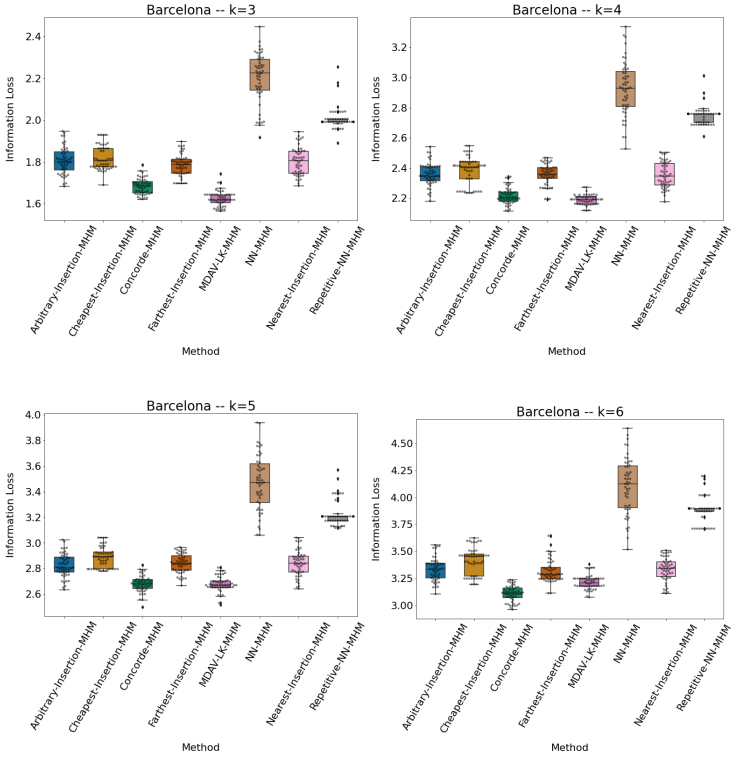

Figure A4. Information Loss variability for each value of k over the Barcelona dataset.

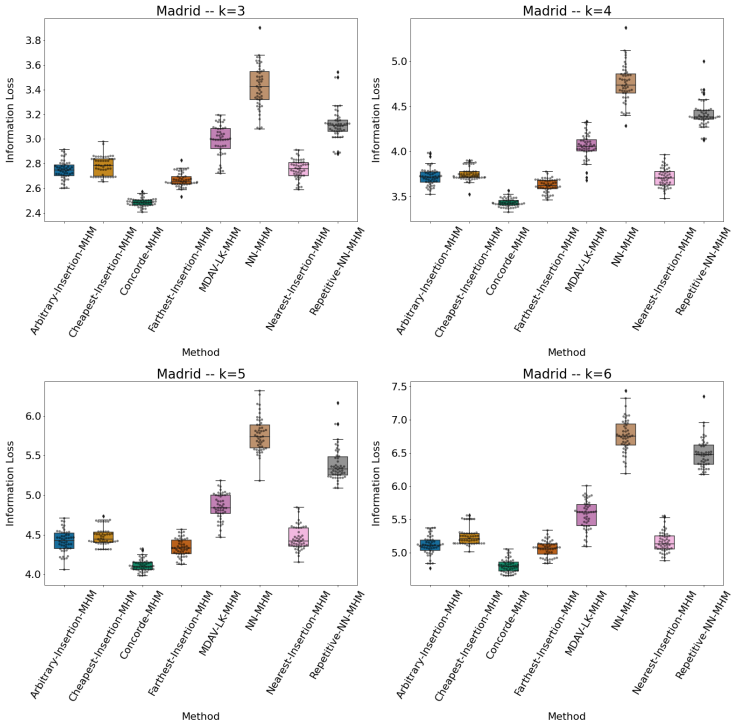

Figure A5. Information Loss variability for each value of k over the Madrid dataset.

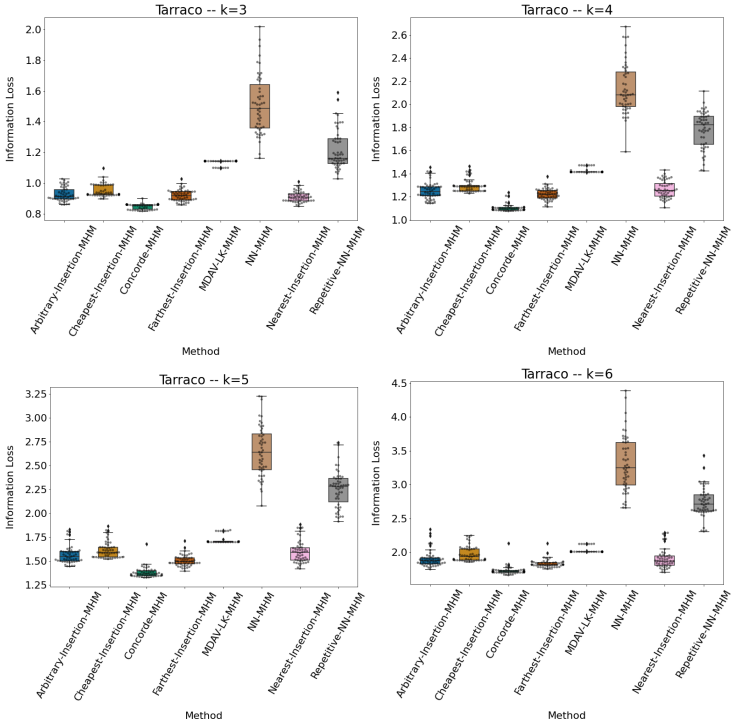

Figure A6. Information Loss variability for each value of k over the Tarraco dataset.

Appendix B. Correlation Analysis between "Path Length" and "SSE"

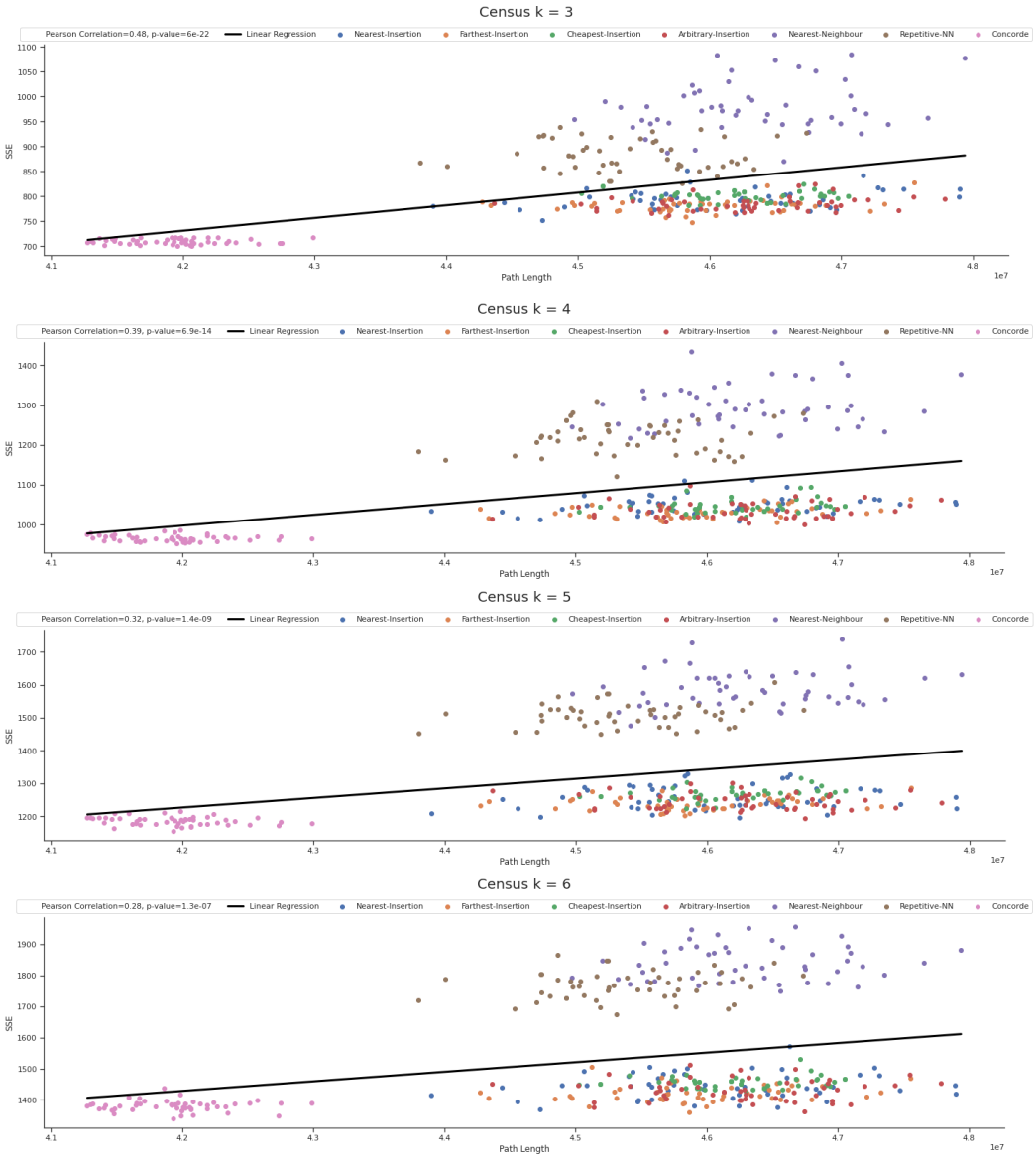

Figure A7. Relation between SSE and Path Length for Census and $k \in \{3, 4, 5, 6\}$.

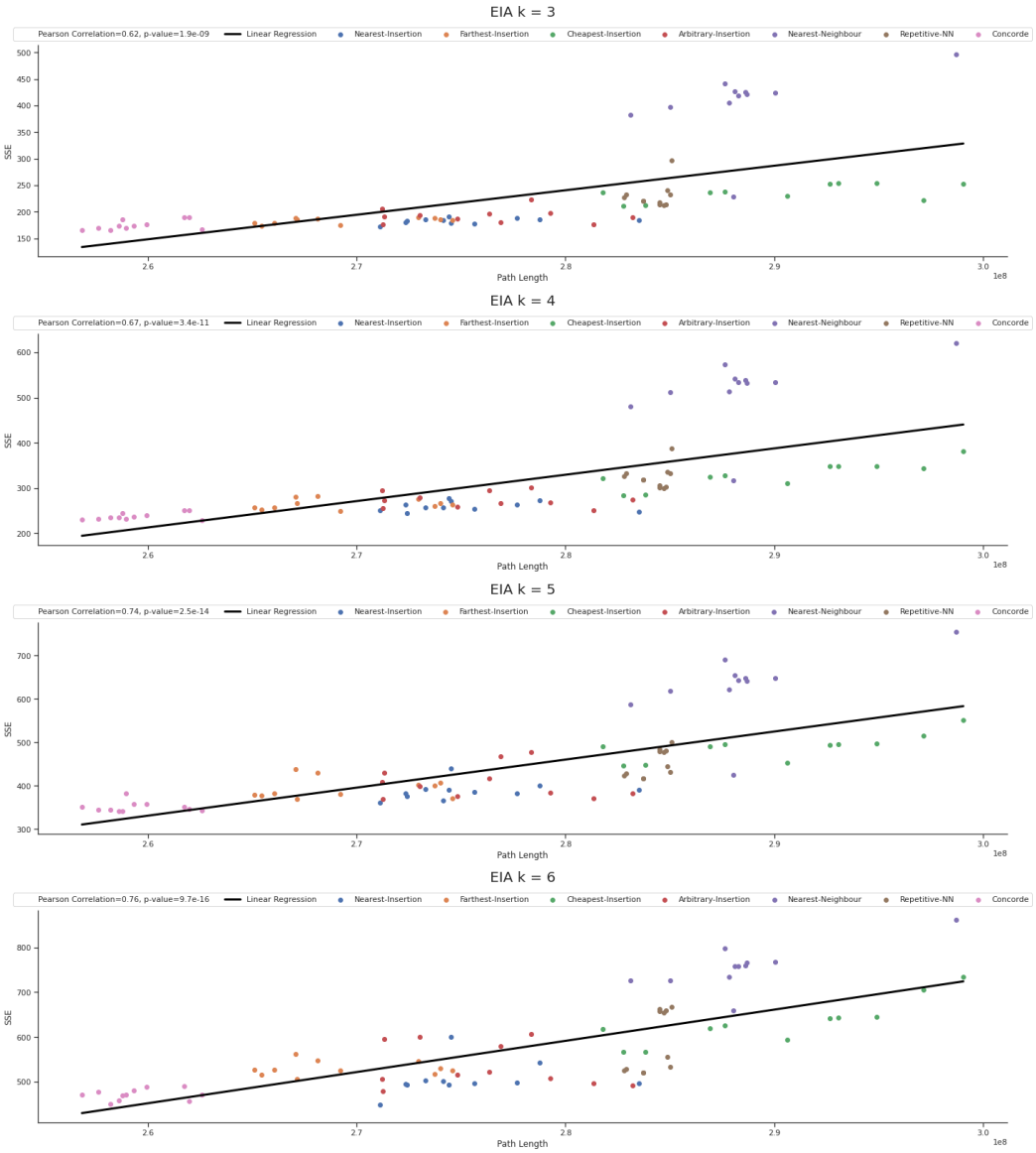

Figure A8. Relation between SSE and Path Length for EIA and $k \in \{3, 4, 5, 6\}$.

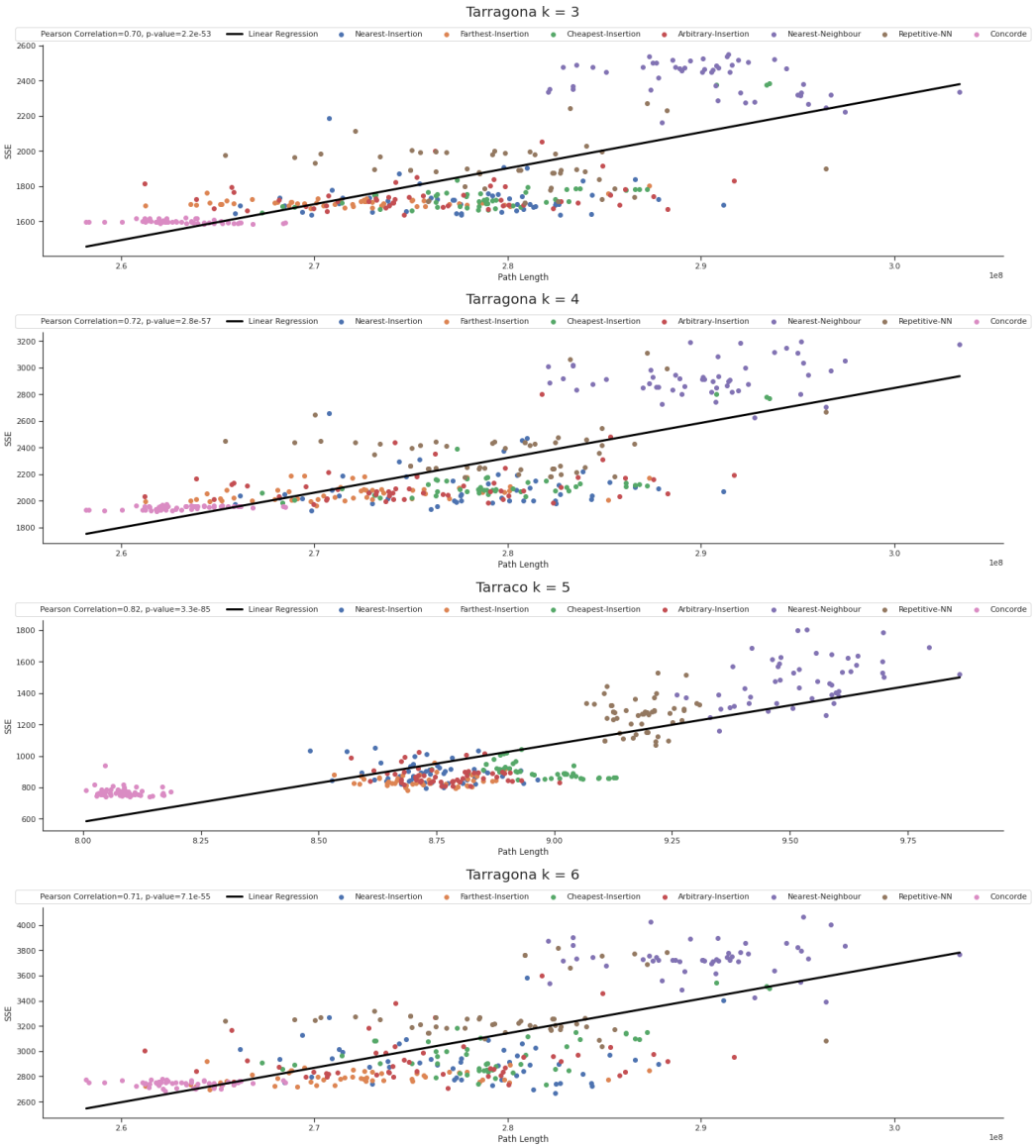

Figure A9. Relation between SSE and Path Length for Tarragona and $k \in \{3, 4, 5, 6\}$.

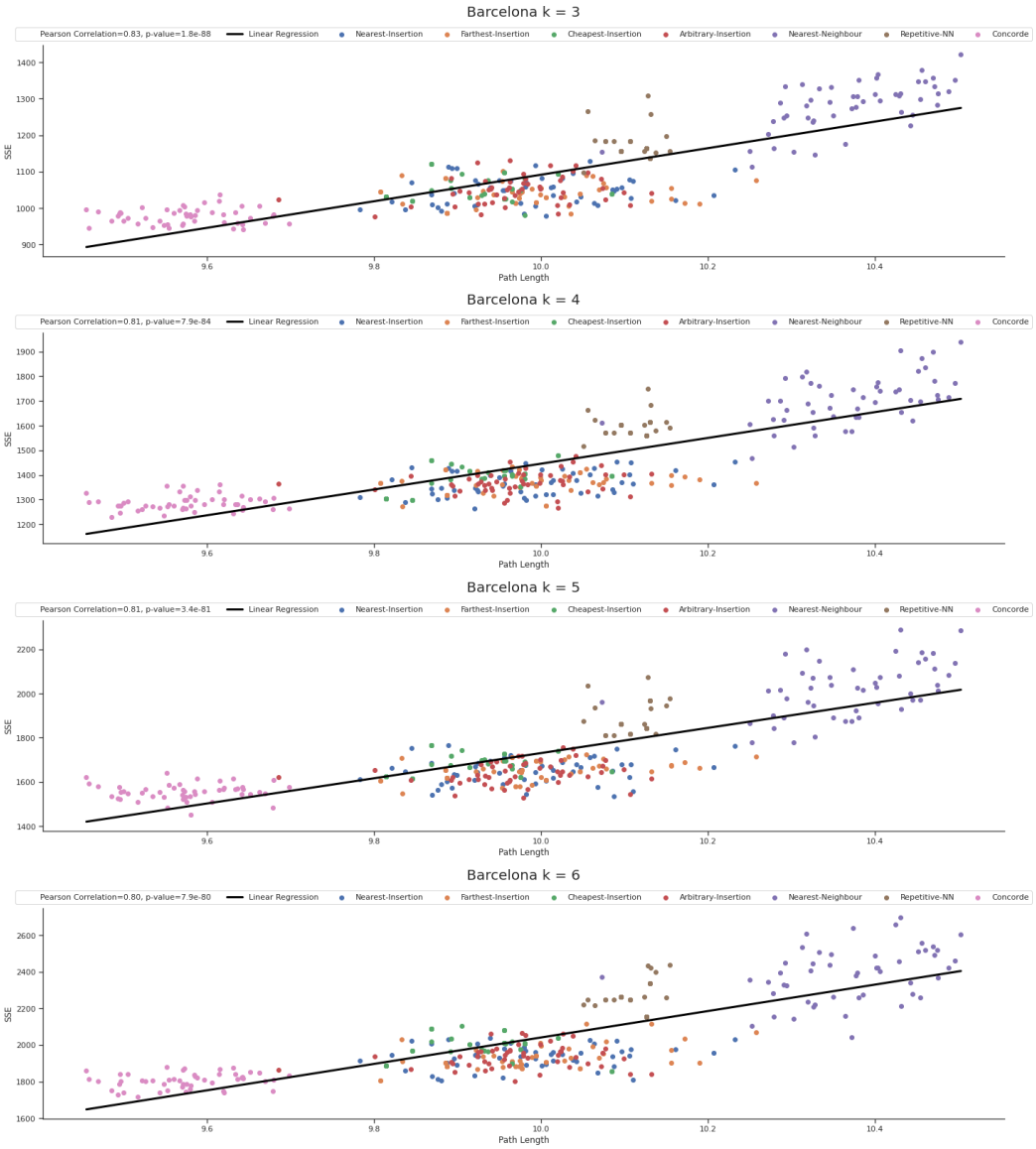

Figure A10. Relation between SSE and Path Length for Barcelona and $k \in \{3, 4, 5, 6\}$.

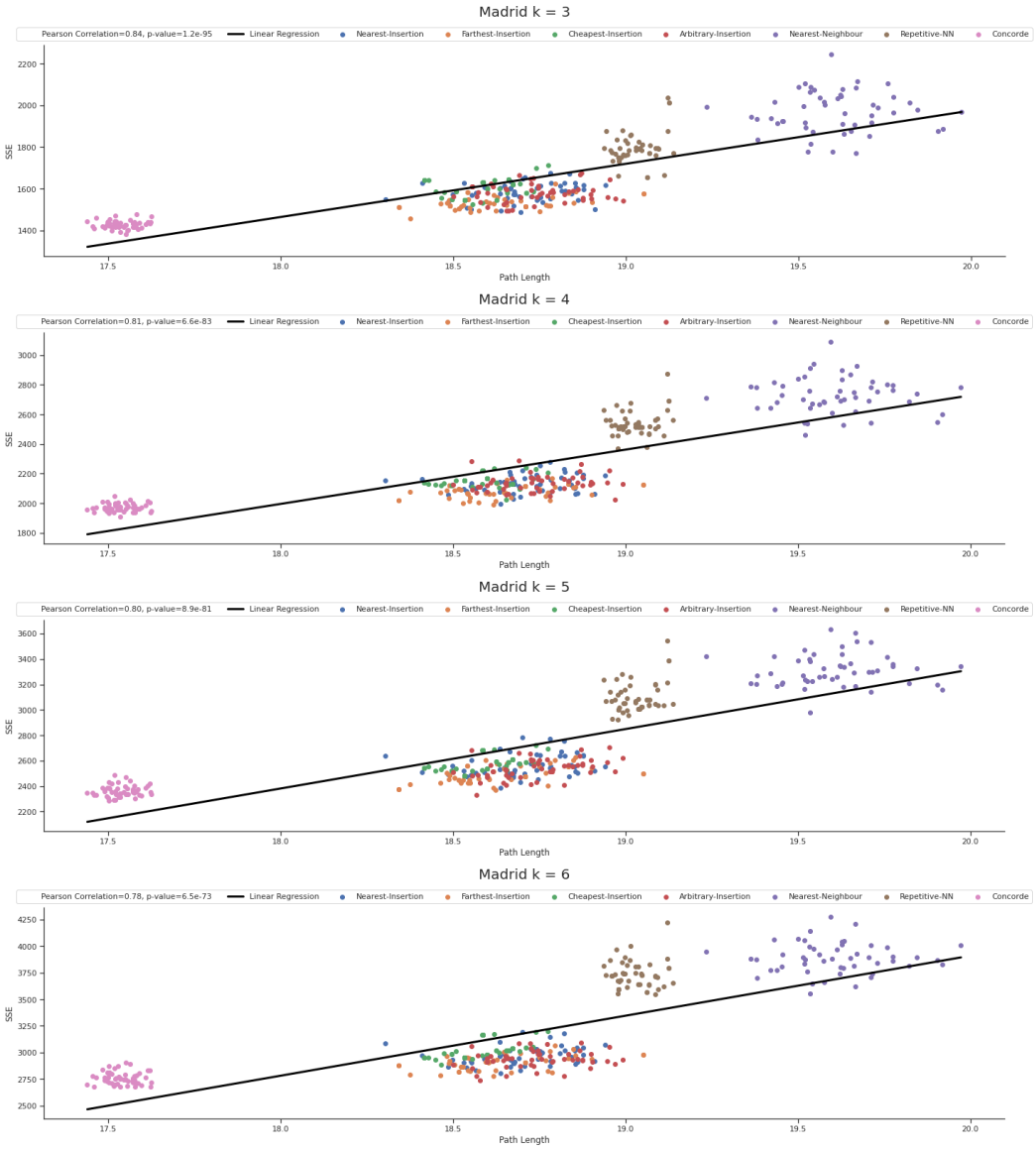

Figure A11. Relation between SSE and Path Length for Madrid and $k \in \{3, 4, 5, 6\}$.

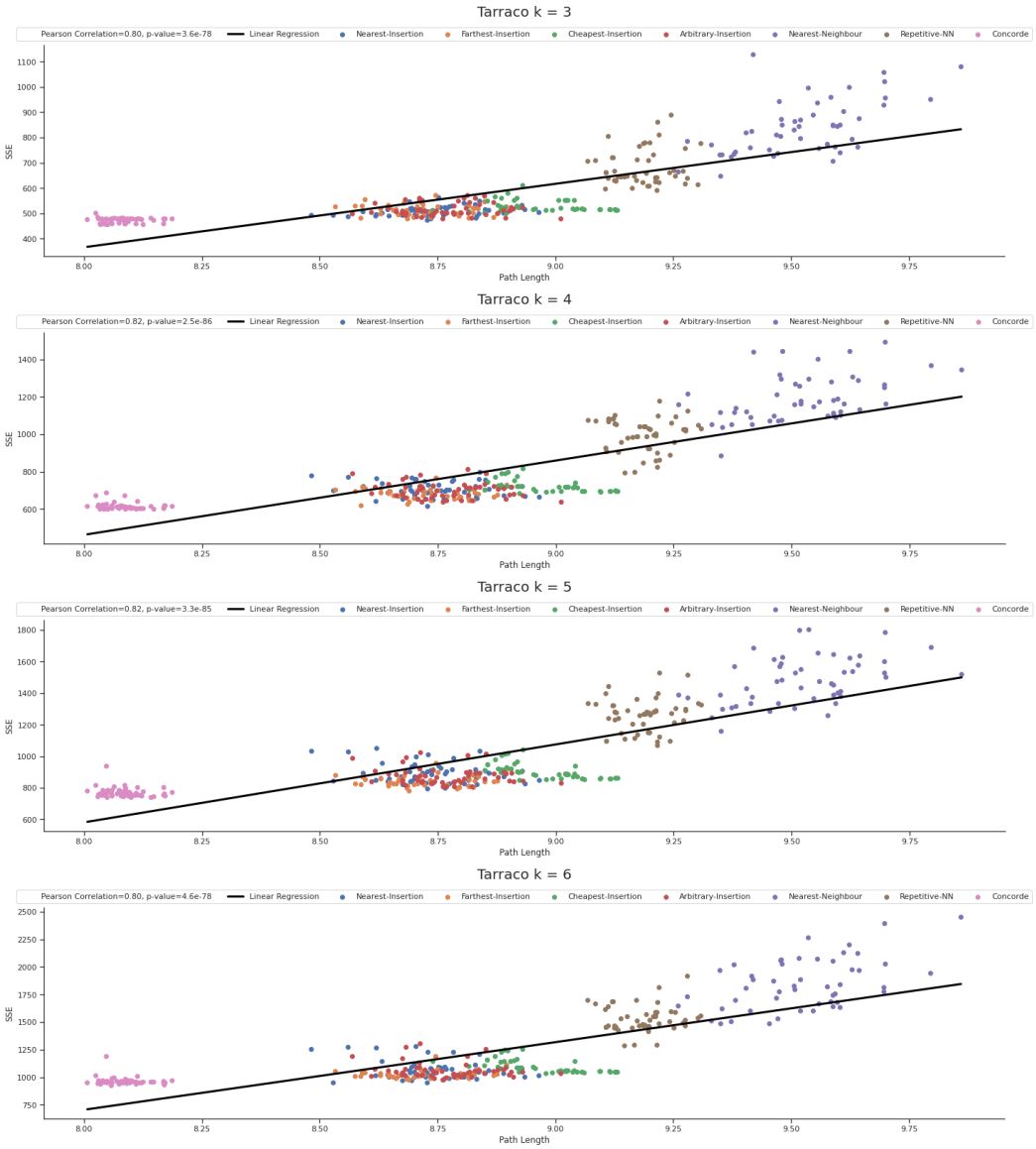

Figure A12. Relation between SSE and Path Length for Tarraco and $k \in \{3, 4, 5, 6\}$.

References

1. Ye, H.; Cheng, X.; Yuan, M.; Xu, L.; Gao, J.; Cheng, C. A survey of security and privacy in big data. In Proccedings of the 2016 16th International Symposium on Communications and Information Technologies (ISCIT), Qingdao, China, 26–28 September 2016; pp. 268–272.
2. Vatsalan, D.; Sehili, Z.; Christen, P.; Rahm, E., Privacy-Preserving Record Linkage for Big Data: Current Approaches and Research Challenges. In *Handbook of Big Data Technologies*; Springer International Publishing: Cham, Switzerlands, 2017; pp. 851–895.
3. Mehmood, A.; Natgunanathan, I.; Xiang, Y.; Hua, G.; Guo, S. Protection of big data privacy. *IEEE Access* **2016**, *4*, 1821–1834. [CrossRef]
4. Barbaro, M.; Zeller, T.; Hansell, S. A face is exposed for AOL searcher no. 4417749. *New York Times*, 9 August 2006.
5. Zigomitros, A.; Casino, F.; Solanas, A.; Patsakis, C. A Survey on Privacy Properties for Data Publishing of Relational Data. *IEEE Access* **2020**, *8*, 51071–51099. [CrossRef]
6. Domingo-Ferrer, J.; Mateo-Sanz, J.M. Practical data-oriented microaggregation for statistical disclosure control. *Knowl. Data Eng. IEEE Trans.* **2002**, *14*, 189–201. [CrossRef]
7. Torra, V. Microaggregation for Categorical Variables: A Median Based Approach. In *Privacy in Statistical Databases*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 162–174.
8. Solanas, A.; Martinez-Balleste, A.; Mateo-Sanz, J.M. Distributed architecture with double-phase microaggregation for the private sharing of biomedical data in mobile health. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 901–910. [CrossRef]
9. Martinez-Balleste, A.; Perez-Martines, P.A.; Solanas, A. The pursuit of citizens' privacy: A privacy-aware smart city is possible. *IEEE Commun. Mag.* **2013**, *51*, 136–141. [CrossRef]

10. Casino, F.; Domingo-Ferrer, J.; Patsakis, C.; Puig, D.; Solanas, A. A k-anonymous approach to privacy preserving collaborative filtering. *J. Comput. Syst. Sci.* **2015**, *81*, 1000–1011. [CrossRef]
11. Domingo-Ferrer, J.; Sebé, F.; Solanas, A. A polynomial-time approximation to optimal multivariate microaggregation. *Comput. Math. Appl.* **2008**, *55*, 714–732. [CrossRef]
12. Hansen, S.L.; Mukherjee, S. A polynomial algorithm for optimal univariate microaggregation. *IEEE Trans. Knowl. Data Eng.* **2003**, *15*, 1043–1044. [CrossRef]
13. Solanas, A.; Martinez, A. VMDAV: A Multivariate Microaggregation With Variable Group Size. In Proceedings of the 17th COMPSTAT Symposium of the IASC, Rome, Italy, 28 August–1 September 2006; pp. 917–925.
14. Shmoys, D.B.; Lenstra, J.; Kan, A.R.; Lawler, E.L. *The Traveling Salesman Problem*; John Wiley & Sons, Incorporated: Hoboken, NJ, USA, 1985; Volume 12.
15. Rosenkrantz, D.J.; Stearns, R.E.; Lewis, P.M., II. An analysis of several heuristics for the traveling salesman problem. *SIAM J. Comput.* **1977**, *6*, 563–581. [CrossRef]
16. Applegate, D.L.; Bixby, R.E.; Chvatal, V.; Cook, W.J. *The Traveling Salesman Problem: A Computational Study*; Princeton University Press: Princeton, NJ, USA, 2006.
17. Domingo-Ferrer, J.; Martinez-Ballesté, A.; Mateo-Sanz, J.; Sebé, F. Efficient multivariate data-oriented microaggregation. *VLDB J.* **2006**, *15*, 355–369. [CrossRef]
18. Laszlo, M.; Mukherjee, S. Minimum spanning tree partitioning algorithm for microaggregation. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 902–911. [CrossRef]
19. Solé, M.; Muntés-Mulero, V.; Nin, J. Efficient microaggregation techniques for large numerical data volumes. *Int. J. Inf. Secur.* **2012**, *11*, 253–267. [CrossRef]
20. Chang, C.; Li, Y.; Huang,W. TFRP: An efficient microaggregation algorithm for statistical disclosure control. *J. Syst. Softw.* **2007**, *80*, 1866–1878. [CrossRef]
21. Yang, G.; Ye, X.; Fang, X.; Wu, R.; Wang, L. Associated attribute-aware differentially private data publishing via microaggregation. *IEEE Access* **2020**, *8*, 79158–79168. [CrossRef]
22. Lin, J.L.; Wen, T.H.; Hsieh, J.C.; Chang, P.C. Density-based microaggregation for statistical disclosure control. *Expert Syst. Appl.* **2010**, *37*, 3256–3263. [CrossRef]
23. Panagiotakis, C.; Tziritas, G. Successive group selection for microaggregation. *IEEE Trans. Knowl. Data Eng.* **2011**, *25*, 1191–1195. [CrossRef]
24. Mortazavi, R.; Jalili, S.; Gohargazi, H. Multivariate microaggregation by iterative optimization. *Appl. Intell.* **2013**, *39*, 529–544. [CrossRef]
25. Mortazavi, R.; Jalili, S. Fast data-oriented microaggregation algorithm for large numerical datasets. *Knowl. Based Syst.* **2014**, *67*, 195–205. [CrossRef]
26. Fayyoumi, E.; Oommen, B.J. A survey on statistical disclosure control and micro-aggregation techniques for secure statistical databases. *Softw. Pract. Exp.* **2010**, *40*, 1161–1188. [CrossRef]
27. Heaton, B.; Mukherjee, S. Record Ordering Heuristics for Disclosure Control through Microaggregation. In Proceedings of the International Conference on Advances in Communication and Information Technology, Amsterdam, The Netherlands, 1–2 December 2011.
28. Templ, M. Statistical disclosure control for microdata using the R-package sdcMicro. *Trans. Data Priv.* **2008**, *1*, 67–85.
29. OpenStreetMap Contributors. 2017. Available online: https://planet.osm.org (accessed on 1 May 2021).
30. Nilsson, C. *Heuristics for the Traveling Salesman Problem*; Technical Report; Linköping University: Linköping, Sweden, 2003. Available online: http://www.ida.liu.se/~TDDB19/reports_2003/htsp.pdf (accessed on 1 May 2021).
31. Morrison, D.R.; Jacobson, S.H.; Sauppe, J.J.; Sewell, E.C. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discret. Optim.* **2016**, *19*, 79–102. [CrossRef]
32. Hahsler, M.; Hornik, K. TSP-Infrastructure for the Traveling Salesperson Problem. *J. Stat. Softw.* **2007**, *23*, 1–21. [CrossRef]
33. Helsgaun, K. An effective implementation of the Lin–Kernighan traveling salesman heuristic. *Eur. J. Oper. Res.* **2000**, *126*, 106–130. [CrossRef]